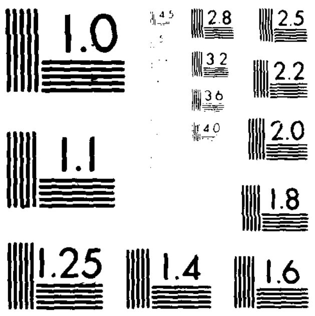
AD-A088 296   NORTH CAROLINA STATE UNIV RALEIGH OPERATIONS RESEARCH   F/G 12/2
ON ESTIMATING THE PROBABILITY DISTRIBUTION FUNCTIONS IN PERT-TY--ETC(U)
JUN 80   B DODIN   DAAG29-79-C-0211
UNCLASSIFIED   NCSU-OR-153-REV   ARO-16352.3-M   NL

1 OF 1
AD
A088296

END
DATE
FILMED
9-80
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

OPERATIONS RESEARCH

NORTH CAROLINA STATE UNIVERSITY AT RALEIGH

ON ESTIMATING THE PROBABILITY DISTRIBUTION FUNCTIONS IN PERT-TYPE NETWORKS

Bajis Dodin*

OR REPORT NO. 153 (Revised) JUNE 1980

Classification Codes: Activity Networks
Statistical Estimation

*Ph.D. candidate, Graduate Program in Operations Research, North Carolina State University, Raleigh, N.C. 27650

This research was partially supported by NSF Grant No. ENG-7813936 and ARO Contract No. DAAG29-79-C-0211.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
16352.3-M ✓

2. GOVT ACCESSION NO.
AD-A088 296

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
ON ESTIMATING THE PROBABILITY DISTRIBUTION FUNCTIONS IN PERT-TYPE NETWORKS

5. TYPE OF REPORT & PERIOD COVERED
Technical

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Bajis Dodin

8. CONTRACT OR GRANT NUMBER(s)
DAAG29-79-C-0211

9. PERFORMING ORGANIZATION NAME AND ADDRESS
North Carolina State University
Raleigh, NC 27650 ✓

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
U. S. Army Research Office
Post Office Box 12211
Research Triangle Park, NC 27709

12. REPORT DATE
Jun 80

13. NUMBER OF PAGES
72

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

estimating
probability
distribution functions
PERT
networks

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This study deals with the problem of approximating the probability distribution function of the project duration in probabilistic activity networks. It describes a procedure that has been developed, programmed and tested, using activity networks of real life projects as well as randomly generated ones. The procedure allows the activity duration to have any of the following distributions: Uniform, Triangular, Normal, Exponential, Gamma, Beta or any discrete distribution presented in a

20. ABSTRACT CONTINUED

finite set of ordered pairs. The computational experience indicates that the resultant probability distribution function (pdf) is very close to the actual pdf, the latter is obtained through extensive Monte Carlo sampling. In fact, computational experience shows that the pdf obtained by Monte Carlo sampling converges to the approximate pdf as the sample size increases. The procedure is programmed in FORTRAN.

ON ESTIMATING THE PROBABILITY DISTRIBUTION FUNCTIONS IN PERT-TYPE NETWORKS

Bajis Dodin
Graduate Program in Operations Research
North Carolina State University
P.O. Box 5511
Raleigh, NC 27650

ABSTRACT

This study deals with the problem of approximating the probability distribution function of the project duration in probabilistic activity networks. It describes a procedure that has been developed, programmed and tested, using activity networks of real life projects as well as randomly generated ones. The procedure allows the activity duration to have any of the following distributions: Uniform, Triangular, Normal, Exponential, Gamma, Beta or any discrete distribution presented in a finite set of ordered pairs. The computational experience indicates that the resultant probability distribution function (pdf) is very close to the actual pdf, the latter is obtained through extensive Monte Carlo sampling. In fact, computational experience shows that the pdf obtained by Monte Carlo sampling converges to the approximate pdf as the sample size increases. The procedure is programmed in FORTRAN and the CPU time for any moderate size project (i.e., G(N,A) $\leq$ G(60,200)) is less than half a minute on AMDAHL V-7.

< or =

I. INTRODUCTION

One of the main difficulties in probabilistic activity networks (PANs) is the determination of the probability distribution function (pdf) of the project completion time. Approximating the probability distribution function becomes very desirable if it can be easily performed and if it results in an estimation close to the actual pdf. Before introducing the proposed approximating procedure, which has these two features, the following definitions and symbols will be used throughout the discussions to follow:

G(N,A): An activity network with N nodes and A arcs. Nodes represent events and arcs represent activities. Node i is connected directly to node j, where i < j, by at most one arc.

|A|: Number of arcs in A.

i: Denotes a node, and is indexed from 1 to N.

$\underline{a}$: Denotes an arc, and is indexed from 1 to |A|.

NS($\underline{a}$): Starting node of activity $\underline{a}$.

NE($\underline{a}$): End node of activity $\underline{a}$.

IN(i): Indegree of node i.

OUT(i): Outdegree of node i.

δ(·): For either node or arc = $\begin{cases} 0 & \text{if the argument variable is "inactive"} \\ 1 & \text{if the argument variable is "active" where an "active" node or arc is one that is retained in the final (irreducible) network.} \end{cases}$

PRE(i) = $\{\underline{a} \mid NE(\underline{a}) = i \text{ and } \delta(\underline{a}) = 1\}$, the set of active arcs that precede node i.

$*$: A convolution operator.

u: Minimum realization of an activity.

v: Maximum realization of an activity.

α: First parameter for the Exponential, Gamma and Beta distributions.

β: The second parameter for the Beta distribution.

m: Mode of a probability distribution function.

$f(x)$: Density function of the random variable X, i.e., $f(x) = dF(x)$.

IML: A list of nodes in the AN whose pdf's are desired, i.e., milestone or key nodes.

$$\text{MCS} = \begin{cases} 0 \text{ if Monte Carlo sampling is not desired} \\ 1 \text{ otherwise} \end{cases}$$

$\bar{F}(\cdot)$: The Fast Fourier Transformation of the argument.

$p_X(x) = p(X=x)$, the probability mass of the random variable X and in short is denoted by $p(x)$.

NIN: Number of intervals in the sampled distributions (pdf's obtained by Monte Carlo sampling).

A node, i, is realized at time T_i, which is a random variable whose pdf is denoted by $F(\tau)$, or simply $F(i)$. On the other hand, an activity is denoted by $\underline{a}$, and has a duration $X_{\underline{a}}$ which is also a r.v. whose pdf is denoted by $F(x)$, or simply $F(\underline{a})$. Furthermore, we denote $CF(\cdot)$ the count of discrete values assumed by the r.v.. Hence, the pdf $F(\cdot)$ may be represented by a set of ordered pairs $\{(\tau_m, p(\tau_m))\}$ for nodes and $\{(x_m, p(x_m))\}$ for arcs, $m = 1,2,3,\ldots,CF(\cdot)$. When we wish to refer to either arc or node we write $\{(R_m, p(R_m)\}$. Let NRR denote the <u>desired</u> number of ordered pairs in the above set of ordered pairs.

The approximating procedure consists of the following five consecutive steps:

1 - Generating Random Activity Networks (GRAN)

2 - Discretizing the Continuous Distributions (DISCRT)

3 - Reducing the Network to Its Irreducible Form (SCAN)

4 - Sequentially Approximating the Irreducible AN (APRXMT)

5 - Testing the Accuracy of the Approximate pdf (SIMULT) and (MAVGDV)

A brief discussion of the functions of each step is given below; however, a detailed discussion is given in the subsequent sections. Flowchart 1 gives the outline of the approximating procedure (the driver program).

In the first step, if G(N,A) is not part of the input data, then it is randomly generated from the space of all feasible AN's with the specified N and $|A|$. In such a case GRAN (a random activity network generator) is used and a rule for assigning a pdf to each arc has to be specified. An activity can have one of the following six continuous distributions:

Uniform, Triangular, Normal, Exponential, Gamma, and Beta,

or any discrete distribution presented in a finite set of ordered pairs.

If any $\underline{a} \in A$ has a continuous pdf, then DISCRT is used to approximate such a distribution by a discrete one. Different discretization methods are presented in Section III.

The first step in estimating the pdf of the project completion time is to reduce the AN wherever possible. Two kinds of reductions can occur:

a. Convoluting two activities in series, which gives rise to a new activity. This occurs if there is a node i with

$$IN(i) = OUT(i) = 1,$$

such a node will be deactivated after the convolution operation, i.e., old $\delta(i) = 1$ becomes new $\delta(i) = 0$.

b. Taking the maximum over two activities having the same starting and ending nodes (in parallel).

An efficient search method for effecting such reduction is developed, it is denoted by "SCAN". It reduces the AN to its irreducible form (IAN). Details of SCAN are given in Section IV. The IAN can be used to determine the unique activities; where an activity is "unique" if it is an element of only one path.

If the irreducible network has more than one arc, then sequential approximation, which is the subject of Section V, is used to determine F(N) as well as the pdf of every active node in the IAN. In certain cases other active nodes in the IAN, beside node N, are of interest to the analyst. The program "APRXMT" prints the pdf of each of these nodes in the form of a table as well as a digital plot. Along with the pdf, the mean and the standard deviation are also given. All such nodes are listed in the input data under the symbol IML. If all the nodes of the AN are elements of the set IML then the sequential approximation is applied to the AN without the use of SCAN.

From this introduction we realize that the error in the approximation occurs due to three causes:

1 - Discretization: The error in approximating the continuous pdf by a discrete one can be kept at any desired level by simply choosing the proper spacing, Δ, in DISCRT.

2 - Independence of the Nodes: In the sequential approximation we assume that the nodes are independent, where, in fact, some nodes may be dependent. If the pdf of each arc in the AN is approximated by a normal distribution, and we are interested in characterizing the pdf of the project completion time by its first four moments, then we can use the results

developed by C. Clark [1]. However, in this report we are interested in approximating the total pdf of the project completion time, not only some of its moments, when the pdf of each activity may not be normal. This is why we chose not to follow Clark's approach, though we recognize its usefulness in the special circumstances to which it is applicable.

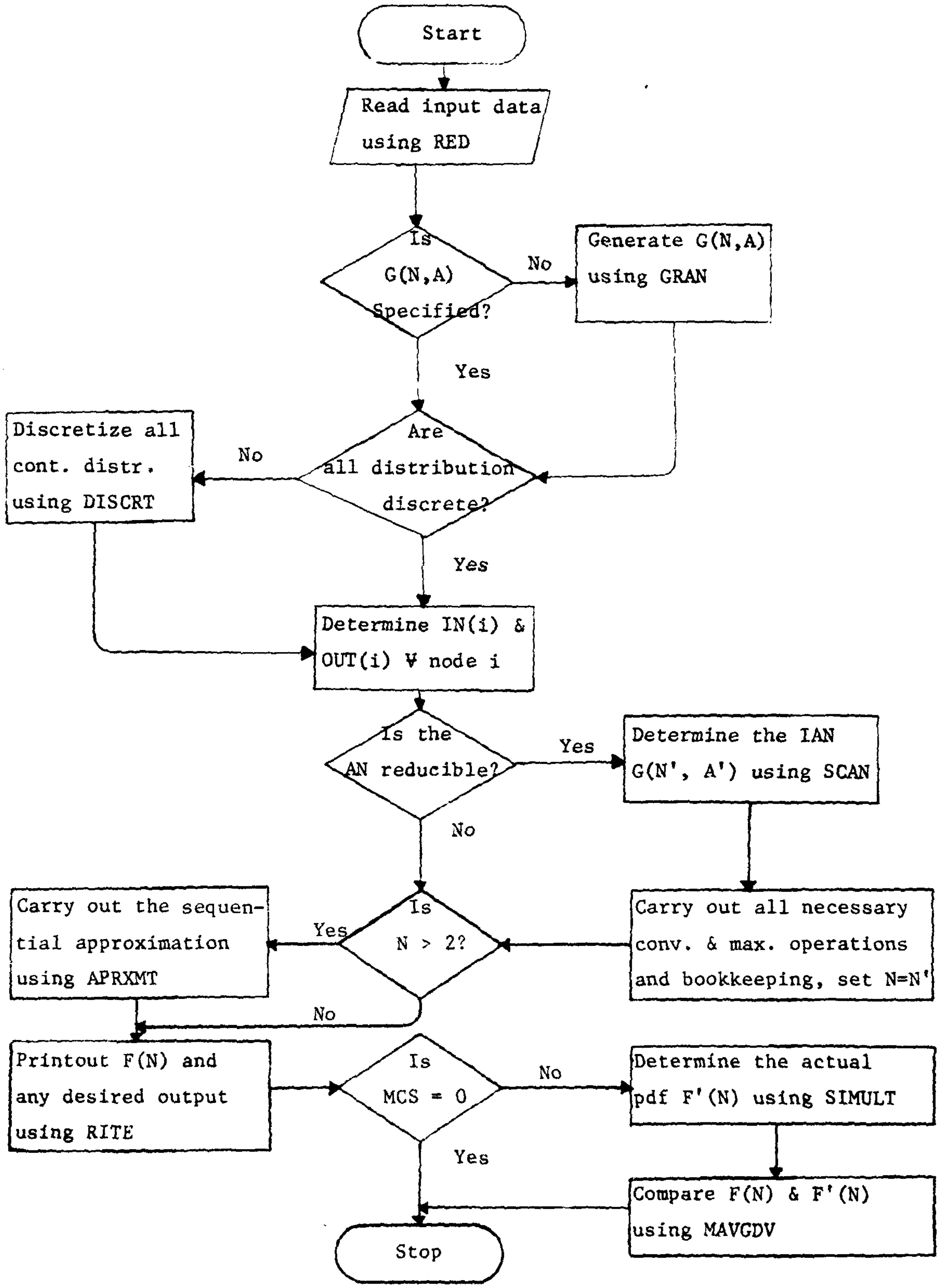
3 - Reducing the Dimension of F(i): In the sequential approximation if CF(i) > NRR, then F(i) is approximated by $F^*(i)$ which has only NRR ordered pairs. The error caused by this approximation can be controlled by choosing large NRR. However, for practical purposes NRR can be chosen to be between 20 and 30; otherwise, the program may run into storage problems, especially if N and $|A|$ are large. If NRR is not binding, then all sets F(i) represent the exact pdf's if all arc pdf's were discrete from the outset; otherwise, the error in F(i) is limited to what is caused by the first two factors.

To be able to measure the error in approximating F(N), the actual F(N) is obtained by extensive Monte Carlo sampling of the original AN using the actual distribution functions. Then the maximum absolute deviations and the average absolute deviations between the two distributions are determined. Also, the actual average and standard deviation of the original pdf are compared with the approximate mean and standard deviation. This is done in Section VI through the use of SIMULT and MAVGDV.

The computational experience presented in Section Section VII indicates that the approximate pdf is very close to that obtained by extensive MCS. In fact, this experience shows that as the sample size in MCS increases the above four measures approach the corresponding values obtained by the approximating procedure.

The procedure has been programmed using FORTRAN IV. It is easy to operate and can be used for any size network after making the necessary storage adjustment. Appendix C describes the input requirements. The program was tested for ANs of sizes (N,A) $\leq$ (60,200). The CPU time in all cases was less than thirty seconds on AMDAHL V-7 excluding the MCS time. The program listing may be obtained from:

Graduate Program in Operations Research
N.C. State University
P.O. Box 5511
Raleigh, N.C. 27650


Start
Read input data using RED
Is G(N,A) Specified?
No
Generate G(N,A) using GRAN
Yes
Are all distribution discrete?
No
Discretize all cont. distr. using DISCRT
Yes
Determine IN(i) & OUT(i) ∀ node i
Is the AN reducible?
Yes
Determine the IAN G(N', A') using SCAN
No
Is N > 2?
Yes
Carry out the sequential approximation using APRXMT
Carry out all necessary conv. & max. operations and bookkeeping, set N=N'
No
Printout F(N) and any desired output using RITE
Is MCS = 0
No
Determine the actual pdf F'(N) using SIMULT
Yes
Compare F(N) & F'(N) using MAVGDV
Stop

Flowchart 1

The Main Algorithm of PDF Approximation

(The Driver)

II. GENERATING RANDOM ACTIVITY NETWORKS

The activity network G(N,A) is either given a priori, which can be an acyclic network of a real life project, or is specified by only N and $|A|$. In the latter case it is desired to generate a feasible acyclic network and to assign a pdf to each activity.

Any procedure used to generate G(N,A) must guarantee that such a network has equal probability to be chosen from the set of all feasible AN's with the specified number of N nodes and $|A|$ arcs. Either of the following two procedures, due to Herroelen [5], can be used to guarantee the complete randomization of G(N,A).

1 - Deletion Method: It starts with a completely connected acyclic network, i.e., the upper triangle of the adjacency matrix is filled with ones. Hence we start the process with N(N-1)/2 arcs in hand, and it is desired to delete

$$K = N(N-1)/2 - |A|$$

arcs subject to the constraints

i) The generated network is feasible, i.e.,

$$\text{IN}(i) \geq 1 \quad \text{for all} \quad i \neq 1$$
$$\text{OUT}(i) \geq 1 \quad \text{for all} \quad i \neq N$$

ii) The generated network is completely randomized, i.e., all networks possessing this count of N and $|A|$ are equally probable.

The Deletion Method does just that. For the sake of completeness, the theory is presented in Appendix A. The method proceeds as follows:

a. Let $OUT(i) = N - i$ for all $i \neq N$

$IN(i) = i - 1$ for all $i \neq 1$

and set $L = 0$

b. Generate a random number, denoted by r_1, where $r_1 \sim U(0,1)$ then let $j = \left\lfloor N + 1/2 - \sqrt{N(N-1)r_1 + 1/4} \right\rfloor$ (1)

c. If $OUT(j) = 1$ or $IN(i) = 1$ for all $i > j$ go to b; otherwise continue.

d. Generate another random number, denoted by r_2, where $r_2 \sim U(0,1)$ and let $k = \left\lfloor j + 1 + r_2(N-j) \right\rfloor$

e. If $IN(k) = 1$ go to d; otherwise arc $\underline{a} = (j,k)$ is deleted, i.e., $\delta(\underline{a}) = 0$. Update $IN(k)$ and $OUT(j)$ and put $L = L + 1$.

f. If $L < K$ go to b; otherwise a completely randomized $G(N,A)$ is in hand and the process stops.

2 - Addition Method: This is the reverse process; it starts with the adjacency matrix filled with zeroes except for

$$\delta(1,2) = \delta(N-1, N) = 1,$$

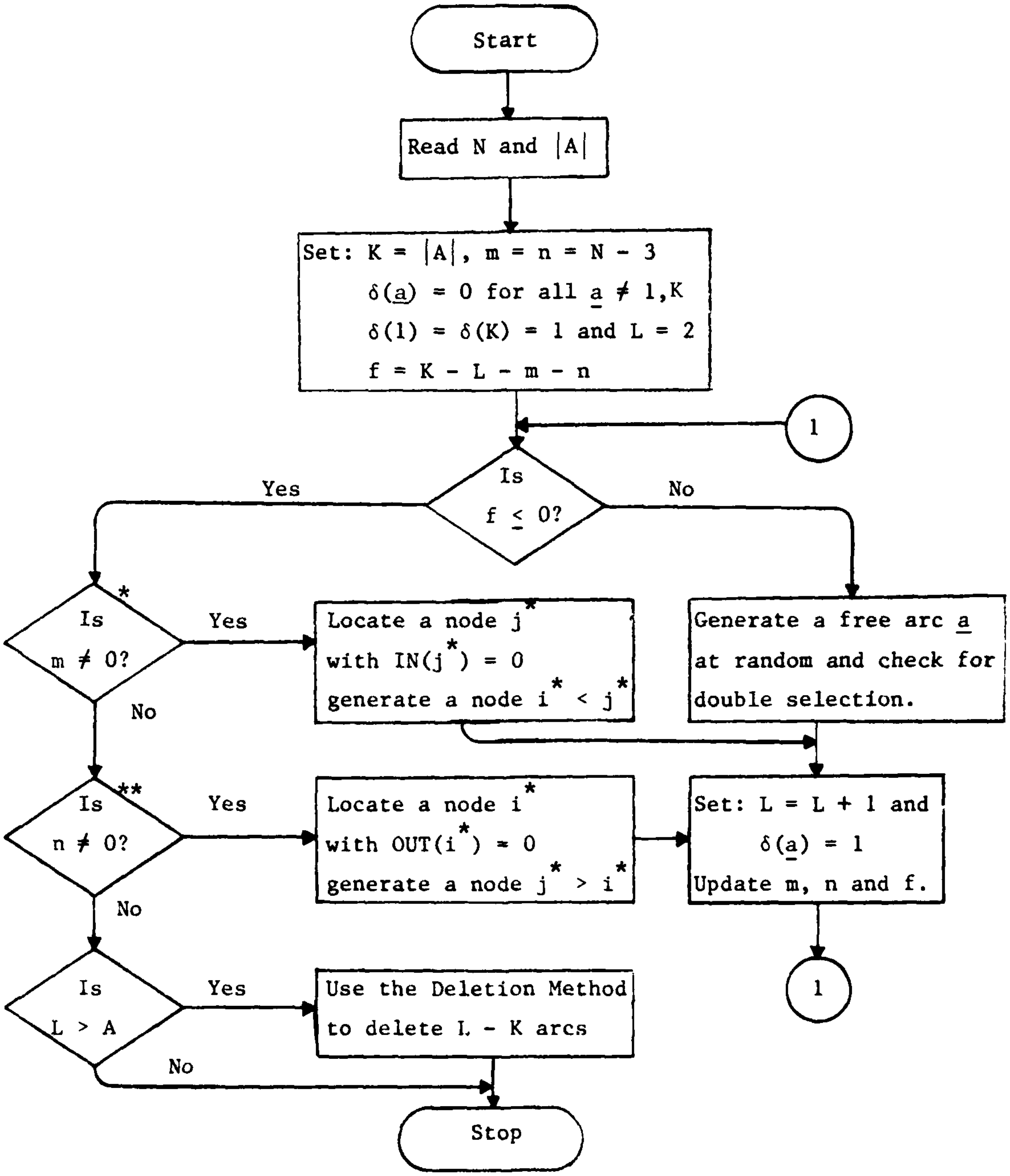
which guarantees one start and one terminal node. The Addition Method then proceeds to generate the remaining $|A| - 2$ arcs subject to constraints (i) and (ii) above. Unfortunately, the Addition Method as developed by Herroelen [5] may generate more than $|A| - 2$ arcs; it may generate an extra M arcs where $0 \leq M < N-3$, especially if $|A| < 2N-4$. In Appendix A, where the theory of the addition method is presented, we dwell more on this problem.

The Addition Method deals with two sets of arcs: the first set has the "feasibility arcs" which range from $(N-3)$ to $(2N-6)$ arcs, and the second set has the remainder of the arcs (if it is nonempty). The second set will be denoted by "free arcs". The Addition Method generates as many as possible of the free arcs in a random fashion which (as shown in Appendix A) may increase as more nodes become feasible. The procedure keeps track of the infeasible nodes, i.e., nodes with zero indegree or zero outdegree (excluding, of course, nodes 1 and N). If the number of free arcs is reduced to zero, then the procedure generates the arcs necessary for feasibility; this step may cause the generation of more than $|A|$ arcs. This problem is solved by using the Addition Method to delete the M extra arcs at random without violating the feasibility of the AN. The Addition Method is summarized in Flowchart 2.

Obviously, either method can be used to generate G(N,A); however, if the network is dense, then the Deletion Method may be preferred since less arcs are to be deleted than added. The Deletion Method is used if

$$|A| \geq N(N-1)/4 \ ,$$

otherwise the Addition Method is used. Unfortunately, the Addition Method is not as efficient as the Deletion Method, and, in fact, is harder to program. Tests of both procedures for large N and $|A|$ proved the validity of the above rule. Both methods are used in the approximating procedure and access to each method is possible by the automatic use of the above rule.



*m: number of non-receiving nodes, *i.e.*, nodes with zero indegree.

**n: number of non-emitting nodes, *i.e.*, nodes with zero outdegree.

Flowchart 2

The Addition Method

III. DISCRETIZING CONTINUOUS DISTRIBUTIONS

Determining or approximating the pdf of the project duration depends on the pdf of each activity in G(N,A). For example, in Figure 1, the r.v.

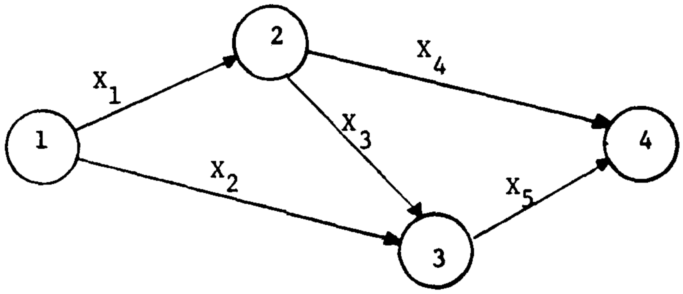


Figure 1

An Irreducible Activity Network

T_4 is a function of all the activities, as shown in Equation (2).

$$T_4 = \mathrm{Max}\{X_1 + X_4, X_1 + X_3 + X_5, X_2 + X_5\} \tag{2}$$

Determining $F(\tau)$ may not be a simple matter especially if some (or all) of the activities have different continuous distributions. Other networks may be more complicated. The determination of $F(\tau)$ is made easier if activities 1 through 5 have discrete distributions; in such a case the digital computer can be used to determine, or approximate, $F(\tau)$

The first step in the approximating procedure is to discretize all the continuous distributions in the AN. This is done by determining a set of ordered pairs denoting $F(\underline{a})$. The cardinality of $CF(\underline{a})$ depends on the desired accuracy of the discretization. Using the closeness to the exact values of the first five moments as a criterion to determine the count of points in the pdf F($\underline{a}$), denoted by CF($\underline{a}$); three methods have been proposed and tried. The most efficient in terms of accuracy and computer time is a hybrid of methods 2 and 3 described below. These three methods are:

1 - The 2m Method: If we decide that CF($\underline{a}$) = m for any activity $\underline{a}$ with continuous distribution, then from the definition of F($\underline{a}$) we have 2m unknowns: m realizations and the corresponding m probabilities. The first 2m moments of the continuous distribution can be used to construct the following system of 2m nonlinear equations:

$$\sum_{k=1}^{m} x_k^n p(x_k) = e_n, \quad \text{for} \quad n = 0,1,2,\ldots,2m-1$$

where

$$e_n = E(X^n) = \int_{-\infty}^{\infty} x^n dF(x)dx, \text{ the } n^{th} \text{ moment.}$$

In a matrix form we have:

$$VP = E$$

where V is the Vandermonde matrix of dimension 2mxm and P is the probability vector with m components, and E is the vector of the 2m moments. Two methods have been tried to solve this system of nonlinear equations, but unfortunately neither succeeded for m > 8. These methods are presented in Appendix B for completeness.

2 - Using Equal Distances: Based on the distribution of the activity, the minimum and maximum realization values u and v can be determined; then by the use of an appropriate spacing Δ, depending on the desired accuracy,

the range (v-u) can be subdivided into equal intervals. In this case

$$x_k = u + \Delta(k-1) \; \forall \; k = 1,2,3,\ldots,m \text{ where } m = \left\lceil \frac{v-u}{\Delta} \right\rceil$$

As a rule in this study the minimum and maximum realizations are determined such that

$$p(X<u) = p(X>v) = 0.0005.$$

The corresponding probabilities are determined according to

$$p(x_k) = \int_{x_k-\Delta/2}^{x_k+\Delta/2} dF(x)dx \quad \text{for each} \quad k = 2,3,4,\ldots,m-1,$$

and $p(x_1) = p(u) = \int_{-\infty}^{u+\Delta/2} dF(x)dx$ and $p(x_m) = p(v) = \int_{v-\Delta/2}^{\infty} dF(x)dx.$ For a small Δ (large m) the determination of the probability can be approximated by $p(y_k) = \Delta\, f(y_k)$ for each $k = 1,2,\ldots,m$ where y_k is the center of the k^{th} interval, i.e., $y_k = u + \Delta(k-1) + \Delta/2$, and $f(y) = dF(y)$. This approach, as it appears in Figure 2, treats all points in the range of the r.v. in a uniform fashion, i.e., we partition the range into equal distances. This makes the discretization suitable for the use of the Fast Fourier Transformation (FFT) method in the successive approximation discussed in Section V. It is also very convenient for some distributions such as the uniform, and the triangular distributions, and some other distributions when their skewness or peaks are not very acute. If sharp peaks are present, such as the case in the exponential distribution with large parameter α, or the normal distribution with small σ, then very small values of Δ are used to minimize the errors of approximation. This drawback led to the use of the following alternative.

3 - Using Equal Probabilities: Here, again, u and v are determined in the same way as in Method 2. Then F($\underline{a}$) is determined according to:

$$\Delta = p(x_k) = 1/m \text{ for a given } m,$$

and

$$x_k = F^{-1}\left(\sum_{j=1}^{k} p(x_j) - \Delta/2\right)$$

using the continuous distribution function. This scheme is suitable for all distributions under consideration. However, it may not be easy (or it can be time consuming) to invert some of the distribution functions. Hence its use is limited to the exponential distribution, where it is needed the most, while the method of equal distances is used for the remaining five distributions. Figures 2 and 3 illustrate methods 2 and 3 respectively for the exponential distribution with parameter $\alpha = 1$. Figure 3 shows that Method 3 responds to the peak of the pdf by taking more realizations, where Figure 2 shows that Method 2 does not respond to peaks.

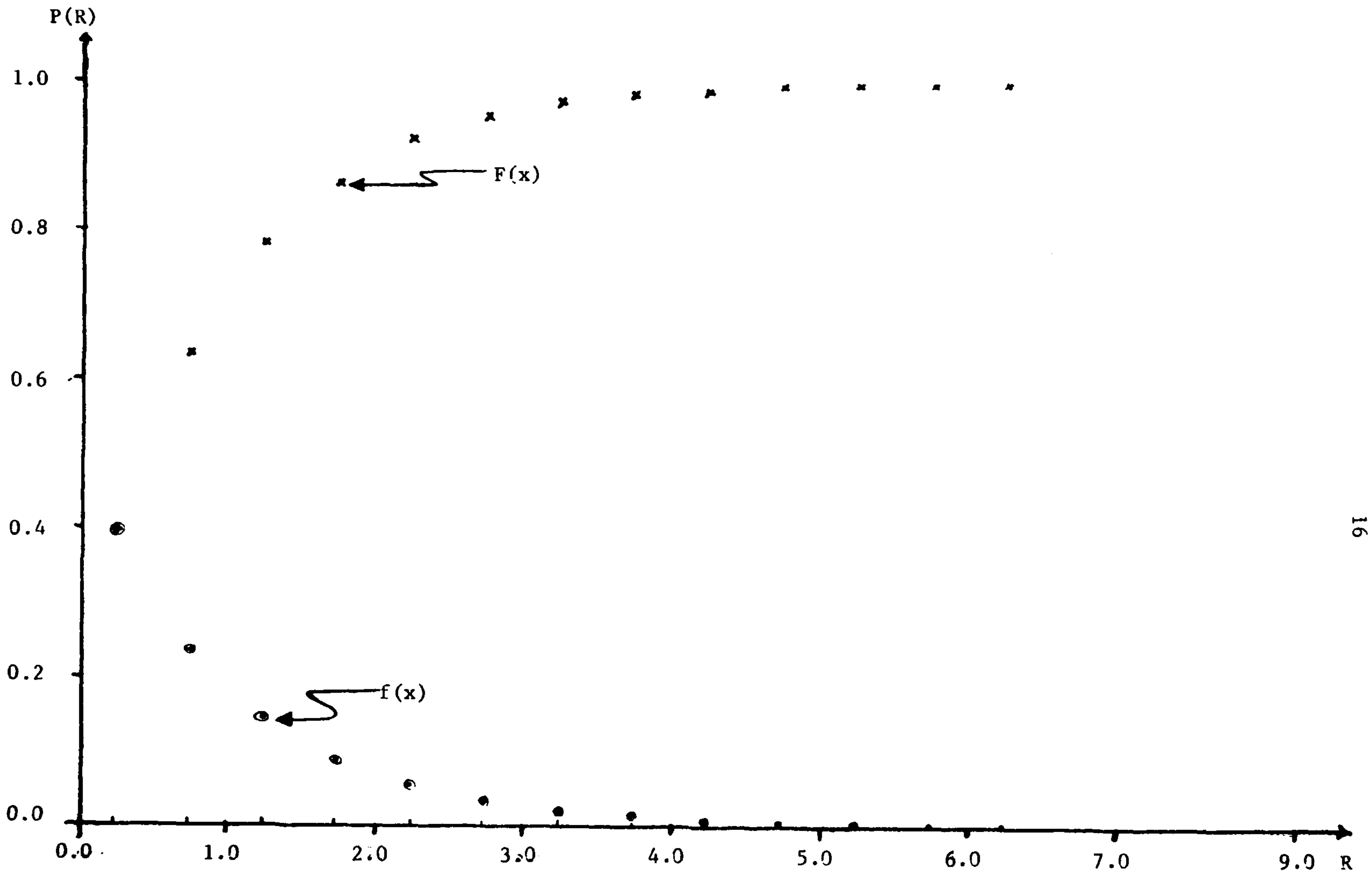


Figure 2

Equal Distance Discretization for the Exponential Distribution

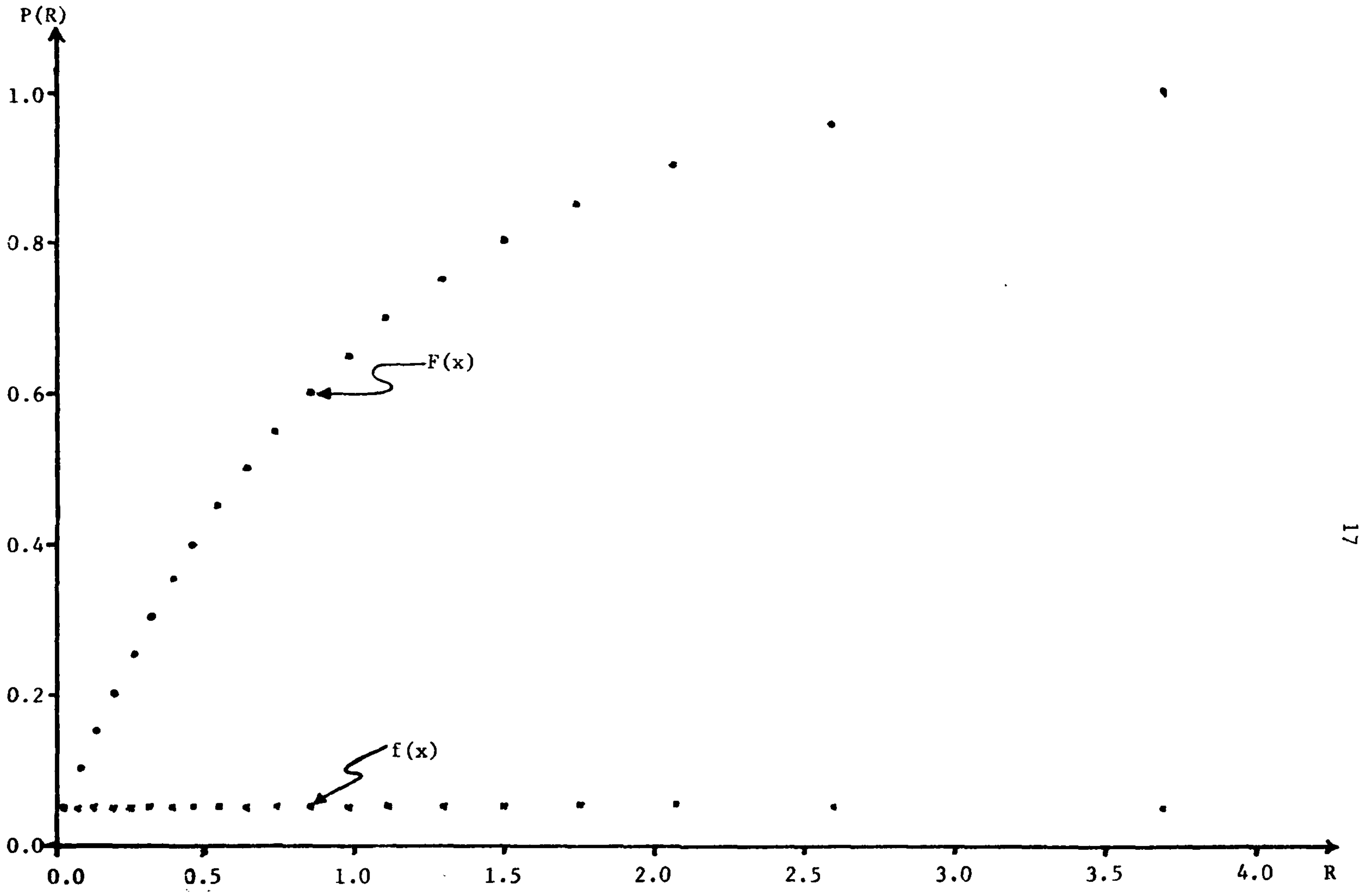


Figure 3

Equal Probability Discretization for the Exponential Distribution

IV. REDUCING THE NETWORK TO ITS IRREDUCIBLE FORM

In an AN it is always possible to combine two arcs in series to form a new arc. Such an operation is accomplished through "convolution". Each convolution operation reduces N and $|A|$ each by one; actually the network gains a new arc, but loses two arcs; this gain and loss are represented by the node and arc indicators. For example, in Figure 4 arc $\underline{a}_1 = (i_1, i_2)$ is convoluted with arc $\underline{a}_2 = (i_2, i_3)$ to give arc $\underline{a}_3 = (i_1, i_3)$.

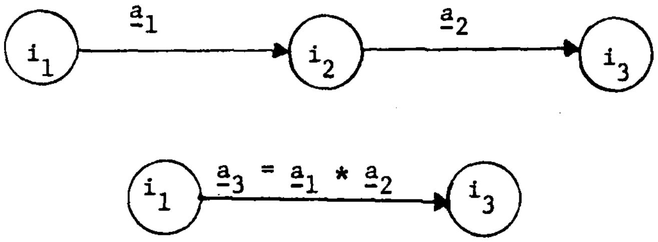


Figure 4

Convolution Operation

This operation introduces the following changes:

$$\delta(i_2) = 0 \quad \text{and} \quad \delta(\underline{a}_1) = 0$$
$$\delta(\underline{a}_2) = 0$$
$$\delta(\underline{a}_3) = 1$$
$$X_3 = X_1 * X_2$$

and the pdf of $\underline{a}_3$ is defined by the set $F(\underline{a}_3) = \{(y, p(y))\}$, where $p_{X_3}(y) = p(X_3 = y) = \sum_{x=0}^{y} p_{X_1}(x)p_{X_2}(y-x)$.

If the nodes i_1 and i_3 in Figure 4 were originally connected with an arc, say $\underline{a}_4$, then the two arcs $\underline{a}_3$ and $\underline{a}_4$ can be "reduced" to one arc $\underline{a}_5$, with duration X_5. Now

$$X_5 = \text{Max}\{X_3, X_4\}$$

and the pdf of X_5 is defined by the set $F(\underline{a}_5) = \{(z, p(z))\}$,where $F_5(z) = F_3(z)\cdot F_4(z)$, and $p_5(z) = F_5(z) - F_5(z^-)$ for a z^- slightly less than z. Figure 5 illustrates this process. Such an operation is called "maximum" operation.

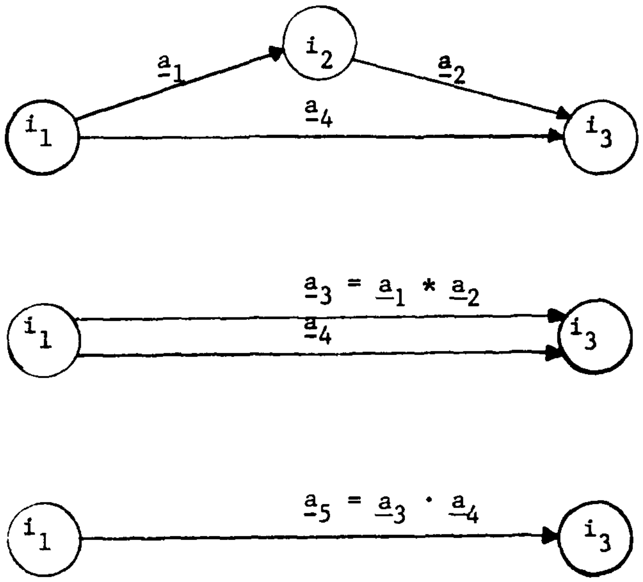

$\underline{a}_1$
i_2
$\underline{a}_2$
i_1
$\underline{a}_4$
i_3
$\underline{a}_3 = \underline{a}_1 * \underline{a}_2$
i_1
$\underline{a}_4$
i_3
$\underline{a}_5 = \underline{a}_3 \cdot \underline{a}_4$
i_1
i_3

Figure 5

Convolution and Maximum Operations

The "maximum" operation reduces the AN by one arc; it sets

$$\delta(\underline{a}_3) = 0$$

$$\delta(\underline{a}_4) = 0$$

and $\delta(\underline{a}_5) = 1$

The reduction process starts with a convolution operation, then a sequence of maximum and convolution operations may follow. The process may start with any of the initial convolutions without fear of not reaching the irreducible form of the AN. Both operations (convolution and maximum) maintain the topological sorting of the AN.

The search for the convolution and maximum operations and the necessary bookkeeping which goes with each operation is carried out by an algorithm called "SCAN", which is based on the following two evident observations:

(i) For any node i, if IN(i) = OUT(i) = 1 then the arcs entering i and emanating from i form a convolution operation.

In some projects node i can be considered a milestone event, i.e., $i \in$ IML, and in such a case $\delta(i) = 1$ throughout the approximating process and the convolution operation may not be carried out.

(ii) For any two arcs $\underline{a} \neq \underline{a}'$ if

$$NS(\underline{a}) = NS(\underline{a}')$$

$$\text{and} \quad NE(\underline{a}) = NE(\underline{a}')$$

then $\underline{a}$ and $\underline{a}'$ form a maximum operation.

At the initial step of the reduction process there does not exist any maximum operation since there is at most one arc connecting any two nodes i and j directly, where $i < j$. The condition for the maximum operation develops after the occurrence of at least one convolution operation; hence, we initially check for a convolution operation using the first observation and the following result which can be proved by induction:

Assertion 1: In any AN if IN(i) + OUT(i) > 2 for all nodes $i \neq 1$ or N, then the AN is irreducible.

Therefore, if IN(i) + OUT(i) > 2 for all $i \neq 1$ or N, then the process moves on to the sequential approximation. In this case the calculations are limited to only N-2 additions and comparisons. However, if there is a node $i \neq 1$ or N such that

$$IN(i) + OUT(i) = 2,$$

then a convolution operation is possible which, if carried out, might give rise to maximum and convolution operations; then the search for both operations becomes necessary. The search procedure based on observations (i) and (ii) above is presented in Flowchart 3. This discussion leads to the following result:

Assertion 2: For any node $i \neq 1$ or N in an irreducible activity network

$$IN(i) + OUT(i) \geq 3 .$$

SCAN was programmed and tested using networks generated by GRAN as well as networks of actual projects.

If the network is reducible then

$$\text{SCAN:} \quad G(N,A) \rightarrow G(N',A')$$

where

$$2 \leq N' < N$$

and

$$1 \leq |A'| < |A|$$

If $N' = 2$ then $|A'| = 1$ and the network is said to be "completely reducible". Then the approximating procedure terminates with the pdf of T_N without resort to the sequential approximation process presented in Section V. This is the ideal situation where no approximation is made except in the discretization stage. Such is the case in the AN of Figure 6 where pdf of T_N reached after 3 convolutions and 2 maximum operations.

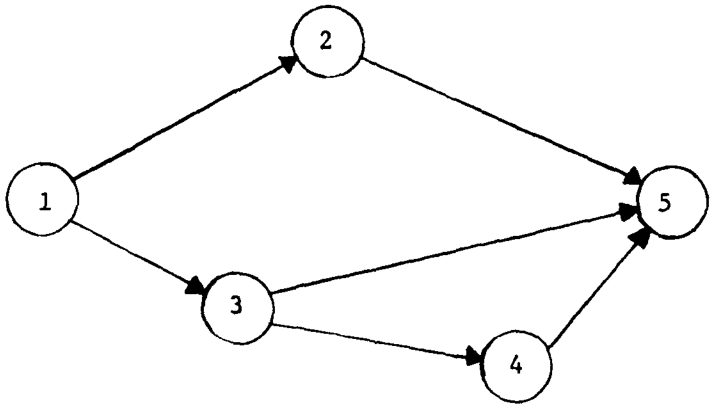
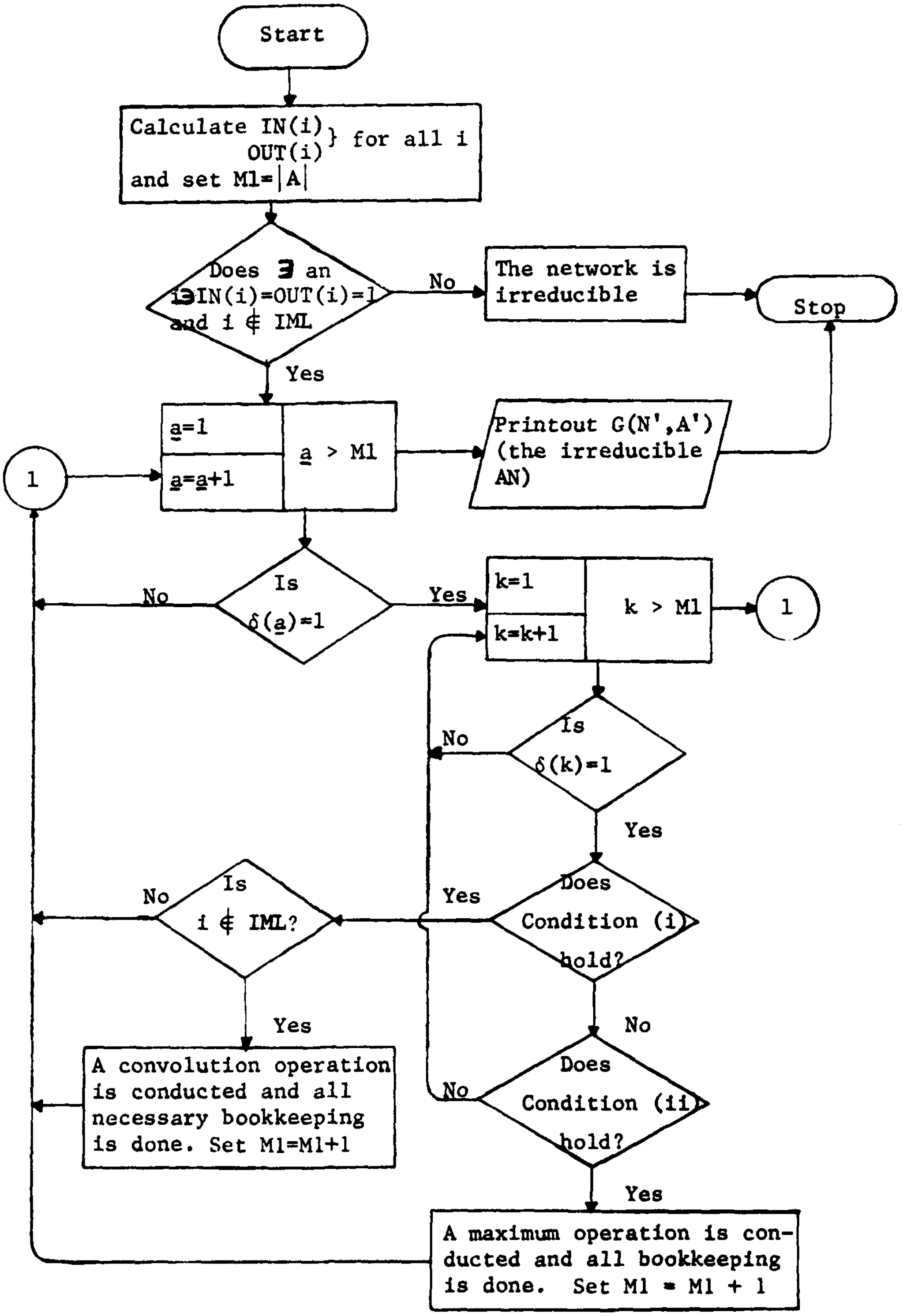


Figure 6

Completely Reducible AN


Start
Calculate IN(i) OUT(i) } for all i and set M1=|A|
Does ∃ an i∋IN(i)=OUT(i)=1 and i ∉ IML
No
The network is irreducible
Stop
Yes
a=1
a=a+1
a > M1
Printout G(N',A') (the irreducible AN)
1
Is δ(a)=1
No
Yes
k=1
k=k+1
k > M1
1
Is δ(k)=1
No
Yes
Is i ∉ IML?
No
Yes
Does Condition (i) hold?
No
Yes
A convolution operation is conducted and all necessary bookkeeping is done. Set M1=M1+1
Does Condition (ii) hold?
No
Yes
A maximum operation is conducted and all bookkeeping is done. Set M1 = M1 + 1

Flowchart 3

An Algorithm for Determining the Irreducible Network

V. SEQUENTIAL APPROXIMATION OF IRREDUCIBLE ANS

The procedure to be described may be used for any AN with $N > 2$ and $|A| > 1$, reducible or irreducible. However, as it is shown in Flowchart 1, we encer this step of the approximating procedure only with irreducible networks; such a network was the result of the SCAN operation, and is denoted by $G(N', A')$. It is represented in the memory of the processor by the active nodes and arcs, i.e., only nodes and arcs with δ's equal to 1. This designation allows us to maintain the original structure of the AN.

Two of the active nodes are: the starting node, 1, where it has the realization time 0 with probability 1, i.e., $F(1) = \{(0,1)\}$, and the terminal node, N, representing the project completion event. The F(N) is the one we are after. Starting at node 1 we proceed to approximate the pdf of every active node, in increasing order, ending with node N. The name "sequential approximation" is derived from this step. The pdf of each active node is approximated by the following procedure, which is illustrated in Flowchart 4.

Without loss of generality assume the process is at node i; node i is active, hence

$$IN(i) > 1 \text{ or } OUT(i) > 1,\ IN(i) + OUT(i) \geq 3, \text{ or } i \in IML,$$

see Figure 7 for illustration, then:

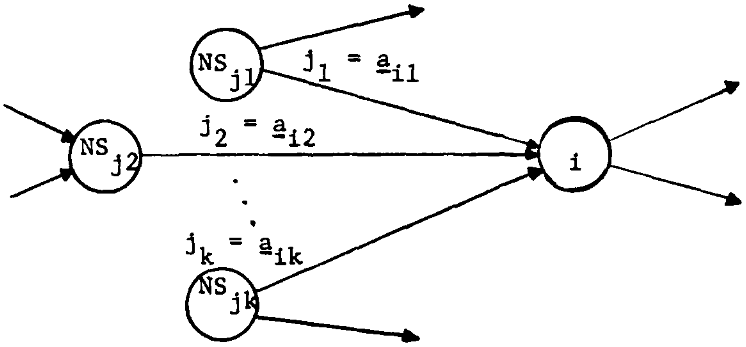

NS_{j1}
$j_1 = \underline{a}_{i1}$
NS_{j2}
$j_2 = \underline{a}_{i2}$
$j_k = \underline{a}_{ik}$
NS_{jk}
i

Figure 7

A Node in an Irreducible AN

1 - Determine the set of active arcs terminating in node i, i.e.,

$$PRE(i) = \{\underline{a} \mid \delta(\underline{a}) = 1 \text{ and } NE(\underline{a}) = i\}.$$

2 - For each $\underline{a} \in PRE(i)$

(a) Convolute $F(\underline{a})$ with $F(NS(\underline{a}))$. Denote this convolution by $F(i_{\underline{a}})$.

(b) If $k = CF(i_{\underline{a}}) > NRR$ then use the operator:

$$APPR: F(i_{\underline{a}}) \rightarrow F'(i_{\underline{a}})$$

where $CF'(i_{\underline{a}}) = NRR$, i.e., approximate the k ordered pairs by only NRR ordered pairs. This is done according to the rules:

(i) The full range of the realization of the project ending at node i is maintained.

(ii) The intermediate k-2 points are mapped into NRR-2 points using the following three steps:

(1) Let $\Delta = (R_{k-1} - R_2)/(NRR-2)$, then we have NRR-2 intervals, each is of width Δ. The n^{th} interval must contain all realizations $R_m \in (R_1 + \Delta(n-1), R_1 + n\Delta]$.

(2) For the realizations in the n^{th} interval let

$$x_n = \sum_m R_m \times p(R_m) \quad \text{and} \quad y_n = \sum_m p(R_m).$$

then

$$(R_n', P(R_n')) = (x_n/y_n, y_n).$$

(3) If the n^{th} interval is empty, then

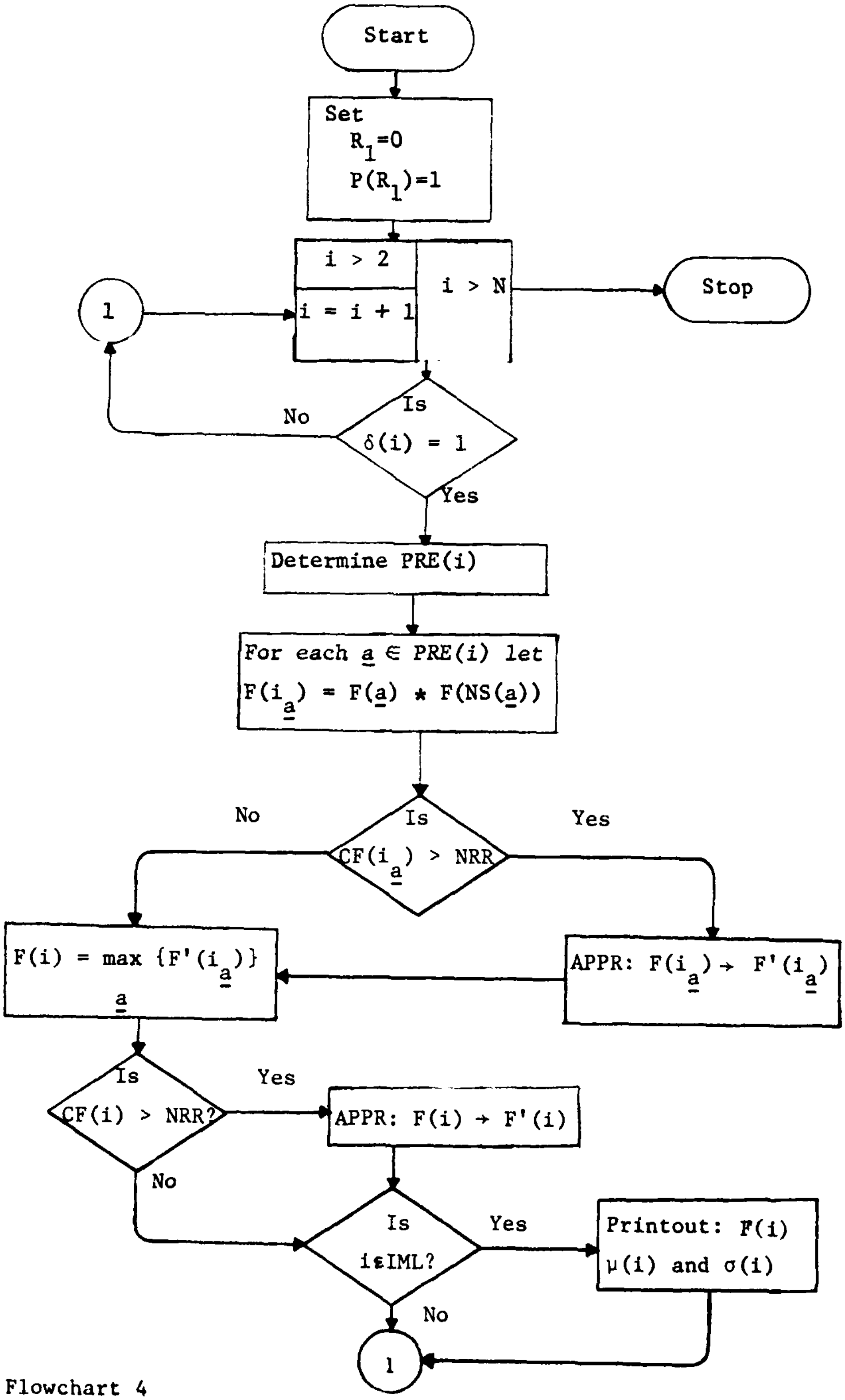
$$(R_n', P(R_n')) = (R_1 + \Delta(n-0.5), 0).$$

3 - $F(i) = \underset{j}{\text{Max}}\{F'(i_{\underline{a}_{ij}})\}$; notice that if APPR is not used then $F'(\cdot) = F(\cdot)$.

This function is separable, hence, to avoid any unexpected escalation in the storage requirements, the maximum can be performed sequentially, then the operator APPR in step (b) above can be used whenever it is necessary. For instance, we let

$$F(i_1, i_2) = \text{Max}\{F'(i_1), F'(i_2)\}$$

then we determine $\text{Max}\{F'(i_3), F(i_1, i_2)\}$ and so on. Hence the process terminates with F(i) where $CF(i) \leq NRR$. The process also determines the mean and the standard deviation, and if $i \in IML$, then F(i) is printed out in the form of a table and digital plot. The process moves on to active node i' immediately succeeding node i. If more than one node succeed node i, then the process starts with the smallest numbered node.



Flowchart 4

Sequential Approximation

The maximum operation used in this step and in SCAN is performed in the usual manner. Perhaps it is best explained by the example with the data in Table 1.

1		2		3 (Max. of 1&2)	
x_1	$p(x_1)$	x_2	$p(x_2)$	x_3	$p(x_3)$
1	1/3	2	1/8	2	2/24
2	1/3	3	3/8	3	6/24
4	1/3	4	3/8	4	13/24
		5	1/8	5	3/24

Table 1

pdf of Two Arcs or an Arc and a Node

$$\begin{aligned} F(3) &= \mathrm{Max}\{F(1),\ F(2)\} \\ &= \{(2,\ 1/24),\ (3,\ 3/24),\ (4,\ 3/24),\ (5,\ 1/24),\ (2,\ 1/24),\ (3,\ 3/24), \\ &\quad (4,\ 3/24),\ (5,\ 1/24),\ (4,\ 1/24),\ (4,\ 3/24),\ (4,\ 3/24),\ (5,\ 1/24)\} \\ &= \{(2,\ 2/24),\ (3,\ 6/24),\ (4,\ 13/24),\ (5,\ 3/24)\} \end{aligned}$$

The complexity of the maximum operation is of $O(\mathrm{NRR})^2$.

The convolution operation can be performed by either the usual formula or through the use of Fast Fourier Transformation (FFT). Testing both methods for different AN's resulted in the preference of the usual formula of convolution. This is due to the following factors:

(i) FFT requires integer subscripts, since for a vector

$$c = (c_o,\ c_1,\ldots,c_{n-1}) \in E_n$$

$$\bar{F}(c_j) = \sum_{k=0}^{n-1} c_k \exp(2i\pi jk/n),\ \forall\ j = 0,1,2,\ldots,n-1 \text{ where } i^2 = -1.$$

Hence, the realization of all activities has to be transferred into multiples of the greatest common divisor; which is a burdensome requirement by itself and poses storage difficulties, compounded with matching increases in the numbers of calculations. Therefore, before FFT is used to convolute

$$(\underline{c},\ p(\underline{c})) \text{ and } (\underline{d},\ p(\underline{d})),$$

each of the vectors $\underline{c}$ and $\underline{d}$ has to be stretched over the range

$$(0,1,2,\ \ldots,\ 2^K) \text{ where } K = \lceil \log_2(c_{n-1} + d_{n-1}) \rceil .$$

Then, the corresponding probabilities have to be assigned accordingly. The example given below illustrates such a preparatory step.

(ii) The use of FFT to convolute $\underline{c}$ and $\underline{d}$ above, after being prepared, proceeds as follows:

(a) Determine $\overline{F}(p(\underline{c}))$ and $\overline{F}(p(\underline{d}))$ using the definition of $\overline{F}(\cdot)$ in (i) above.

(b) Let $\overline{F}(p(\underline{e}))$ be the pairwise product of $\overline{F}(p(\underline{c}))$ and $\overline{F}(p(\underline{d}))$.

(c) Invert using $\overline{F}^{-1}(\overline{F}(p(\underline{e}))$.

Obviously, the three steps involve many transformations and the complexity of the procedure, at best, is of the $0(2^K \ln 2^K) = 0(n \log n)$, since we defined $K = \ \log_2 n$, see [7].

(iii) After $p(\underline{e})$ is obtained, a reduction transformation may be necessary since the set $\{(e_i,\ p(e_i))\}$ is of length 2^K and many of the probabilities $p(e_i) = 0$, especially at the start and end tails of the set. Also, a common factor may be subtracted from the realizations e_i so that they may start at zero, i.e., use a shifting operator.

The following example, which uses the data of X_1 and X_2 in Table 1, illustrates the use of both methods; usual convolution (definitional form) and the preparatory step of FFT. Step (ii) above was processed on the computer and resulted in a pdf close to the exact pdf presented in Table 2. The deviation is due to the truncations and transformations used in the FFT procedure (step (ii) above).

Using the definition of the convolution operator to X_1 and X_2 of Table 1 gives

$$Y = X_1 * X_2 \quad \text{where}$$

$$p_Y(y) = \sum_{z=0}^{y} p_{X_1}(z) p_{X_2}(y-z)$$

This operation is summarized in Table 2 below

y	p(y)	P(y)
3	1/24	1/24 = .0417
4	3/24 + 1/24	5/24 = .2083
5	3/24 + 3/24	11/24 = .4583
6	1/24 + 3/24 + 1/24	16/24 = .6667
7	1/24 + 3/24	20/24 = .8333
8	3/24	23/24 = .9583
9	1/24	24/24 = 1.000

Table 2

The Convolution of X_1 and X_2 of Table 1

To apply FFT, the vectors X_1 and X_2 have to have the common index $n = (0,1,2,...,16)$, notice that

$$16 = 2^4 = 2^K \quad \text{where} \quad K = \lceil \log_2(4+5) \rceil .$$

and the probability vectors are assigned accordingly. Therefore:

n=	0	1	2	3	4	5	6	7	8	9	10	11	...	16
p(x_1)=	0	1/3	1/3	0	1/3	0	0	0	0	0	0	0	...	0
p(x_2)=	0	0	1/8	3/8	3/8	1/8	0	0	0	0	0	0	...	0
p(x_1)=	0	0	0	.0416	.1665	.2497	.2081	.1665	.1249	.0416	0	0	...	0

Table 3

Convolution Using FFT Method

The value of n can be reduced to 8 by subtracting

$$\mathrm{Min}\{Y\} = \mathrm{Min}\{X_1\} + \mathrm{Min}\{X_2\}$$

$$= 1 + 2 = 3$$

Hence Table 3 can be transferred to Table 4 after dropping all entries with zero probabilities at each end of the Table.

n'	0	1	2	3	4	5	6	7	n = n' + 3
P(x_1')	1/3	1/3	0	1/3	0	0	0	0	$x_1 = x_1' + 1$
P(x_2')	1/8	3/8	3/8	1/8	0	0	0	0	$x_2 = x_2' + 2$
P(x_1')	.0416	.1665	.2497	.2081	.1665	.1249	.0416	0	$x_1 = x_1' + 3$

Table 4

The Reduction of Table 3

VI. TESTING THE ACCURACY OF THE APPROXIMATE PDF

The accuracy of the approximate pdf of the project duraticn, represented by F(N) can be measured by its closeness to the "true" pdf, denoted by F'(N). Such "closeness" is measured either by the maximum value of the absolute deviation of F(N) from F'(N), denoted by MDV, or the average value of the absolute deviations, denoted by ADV.

This Section deals with the problem of determining such maximum and average deviations. It first deals with determining F'(N). This is the subject of the first subsection, where Monte Carlo sampling is used, since obtaining F'(N) analytically is not feasible in the majority of cases. Then, in subsection 2, linear interpolation is used to determine the maximum and average absolute deviations (MDV and ADV).

This segment of the Approximating Procedure is used only as an evaluation tool. Access to this test is possible by setting the parameter MCS = 1 in the input data. Sampling consumes a lot of CPU time, hence in dealing with large AN's, allocation of time should be considered before setting MCS = 1. The following is a discussion of the sampling model called "SIMULT".

1 - Sampling the Activity Network: Monte Carlo sampling of the AN is performed using subroutine SIMULT. It assigns a random number to each arc of the original network, G(N,A), generated from the original pdf of the activity, continuous or discrete. Then SIMULT determines the completion time of the project using the longest path method; this also results in the realization times of all nodes in the AN. These two steps are

repeated for a "satisfactory" number of times. Here the number of samples should be "large" to guarantee that F'(N) is very close to the "true" pdf of the project duration time. For example, Table 5 shows for G(10,15) the improvement in MDV and ADV as the number of samples increases. Hence extensive Monte Carlo sampling is necessary; otherwise the values of MDV and ADV may have to be adjusted to reflect the error in the sampled distribution F'(N). The trend of improvement in the values of MDV and ADV in Table 5 indicates that F(N) may be closer to the "true" pdf than these two measures indicate.

Problem	Dists. Type	No. of MCSs	Comparison of the Approximate pdf with that of MCS					
			Average		S. Deviation			
			APRX.	MCS	APRX.	MCS	MDV	ADV
1	All	100	28.46	27.74	3.808	4.434	0.0556	0.0174
2	All	300	28.46	27.83	3.808	3.772	0.0568	0.0158
3	All	500	28.46	27.80	3.808	3.690	0.0584	0.0159
4	All	750	28.46	28.00	3.808	3.904	0.0338	0.0091
5	All	1000	28.46	28.04	3.808	4.005	0.0274	0.0072

Table 5

Effects of Number of Samples

on the Measures of Performance

The gathered information through sampling is tabulated for each of the critical nodes (elements of IML) or only for node N. Each table has three columns; these are: completion times R_i, corresponding probabilities $p(R_i)$ and the accumulative probabilities $P(R_i)$. The first two columns represent the ordered pairs of the pdf F(N). Table 6 was generated using SIMULT,while Table 7 is the corresponding table using the approximating procedure for the same AN,which is G(10,15) with all the pdf's under consideration are used.

The SIMULT subroutine has the potential of simulating any AN provided that each of its arcs has one of the following distributions:

1 - Uniform

2 - Triangular

3 - Normal

4 - Exponential

5 - Gamma

6 - Beta

7 - Discrete distributions or customer specified distributions, where these two cases are represented by a set of finite ordered pairs.

Each of the distributions is identified to SIMULT by its number in the above listing, denoted by I. Hence I = 1 if the activity has a uniform distribution, and I = 2 if the distribution is triangular, and so on. The distribution identity I associated with activity $\underline{a}$ is denoted by the symbol NDST($\underline{a}$). Each distribution is characterized by the following four parameters - (where the use of each parameter is explained in the discussion of each distribution):

EX(I): a real value representing the mean in some distributions, and in some others it is used to input other parameters (such as the first parameter for Gamma and Beta).

STDX(I): real value representing the standard deviation in some distributions and another parameter in some other distributions.

VMIN(I): minimum value of the random variable $X_{\underline{a}}$, determined by the criterion used in DISCRT.

VMAX(I): as VMIN(I) except it represents the maximum of $X_{\underline{a}}$.

The "true" distribution function represented by the Monte Carlo sampling for the original network.

```
  J   REALIZATION      PROBABILITY     ACC. PROB.
  1    15.9925         .100000E-02     .100000E-02
  2    16.9665         .300000E-02     .400000E-02
  3    18.0322         .300000E-02     .700000E-02
  4    18.8939         .300000E-02     .999999E-02
  5    19.5900         .999999E-02     .200000E-01
  6    20.5030         .150000E-01     .350000E-01
  7    21.3058         .220000E-01     .569999E-01
  8    22.2076         .280000E-01     .849999E-01
  9    22.9867         .399999E-01     .125000
 10    23.9390         .559999E-01     .181000
 11    24.8600         .559999E-01     .237000
 12    25.6529         .799996E-01     .316999
 13    26.5022         .849996E-01     .401999
 14    27.4174         .819996E-01     .483998
 15    28.1880         .799996E-01     .563998
 16    29.1118         .819996E-01     .645998
 17    29.9693         .789996E-01     .724997
 18    30.8674         .699998E-01     .794997
 19    31.6662         .599999E-01     .854997
 20    32.4584         .319999E-01     .886997
 21    33.4634         .349999E-01     .921997
 22    34.2368         .260000E-01     .947997
 23    35.0414         .220000E-01     .969997
 24    36.1395         .999999E-02     .979996
 25    36.9213         .799999E-02     .987996
 26    37.7353         .599999E-02     .993996
 27    38.2952         .200000E-02     .995996
 28    39.4778         .0              .995996
 29    40.7200         .300000E-02     .998996
 30    42.8619         .100000E-02     .999996
```

The average duration of the project = 28.04 and its S.D. = 4.005

Table 6

F(10) Generated by SIMULT

```
  J  RFALIZATION      PROBABILITY     ACC. PROB.
  1    8.75000        .139970E-16     .139970E-16
  2    9.78504        .954873F-17     .235457F-16
  3    10.6420        .359588E-11     .359590E-11
  4    12.6604        .341903E-07     .341939E-07
  5    14.0789        .124399E-05     .127818E-05
  6    15.1314        .110340E-04     .123122E-04
  7    16.4698        .140472F-03     .152785F-03
  8    17.5058        .102961E-05     .153814E-03
  9    18.2941        .215756E-02     .231137E-02
 10    20.4425        .196633F-01     .219747E-01
 11    21.1891        .0              .219747E-01
 12    21.9082        .284878E-01     .504625E-01
 13    23.1433        .539027E-01     .104365
 14    24.4398        .932945E-01     .197660
 15    25.7718        .115894         .313554
 16    27.0051        .118442         .431995
 17    28.4196        .155535         .587530
 18    29.8350        .118626         .706156
 19    31.1364        .933400E-01     .799496
 20    32.3999        .800499F-01     .879546
 21    33.7556        .555444E-01     .935090
 22    35.0372        .313673E-01     .966458
 23    36.2193        .185161E-01     .984974
 24    37.4626        .779667E-02     .992770
 25    38.7723        .573234F-02     .998503
 26    40.2990        .108035E-02     .999583
 27    41.6743        .365955E-03     .999949
 28    43.0373        .214648E-04     .999970
 29    44.5454        .722603F-05     .999978
 30    47.2500        .177579E-07     .999978
```

The average duration of the project = 28.46 and its S.D. = 3.808

Table 7

F(10) Generated by APRXMT

To obtain a random value x corresponding to the random variable X with a given distribution F(x), we obtain a random number $y \sim U(0,1)$ then solve for x using the equation

$$F_X(x) = F(X \leq x) = y.$$

Hence

$$x = F^{-1}(y).$$

This logic is used in the subroutines developed by International Mathematical and Statistical Libraries, Inc. (IMSL) [6]. However, in most cases IMSL generates the random number for the standard distribution, such as the case for standard normal, Beta,... . Let r represent the r.n. obtained by IMSL subroutines. The necessary transformation is made to give us the desired r.v. x. Table 8 has a listing of the distributions and the necessary input data represented by the first five columns of the table. Also Table 8 illustrates the use of the above four parameters. The following is a brief description of each distribution:

1 - Uniform Distribution:

$$f(x) = \begin{cases} \frac{1}{v-u} & \text{if} \quad u \leq x \leq v \\ 0 & \quad\text{otherwise} \end{cases}$$

r: is uniformly distributed between 0 and 1.

$x = u + r(v-u)$

$\mu = (u+v)/2$

$\sigma^2 = (v-u)^2/12$

2 - Triangular Distribution: Let $k_1 = (v-u)(m-u)$

and $k_2 = (v-u)(v-m)$

$$f(x) = \begin{cases} 2(x-u)/k_1 & \text{if} \quad u \leq x \leq m \\ 2(v-x)/k_2 & \text{if} \quad m \leq x \leq v \\ 0 & \text{otherwise} \end{cases}$$

r: is between 0 and 1, and has a uniform distribution

$$x = \begin{cases} u + \sqrt{rk_1} & \text{if} \quad 0 \leq r \leq y \quad \text{where} \quad 0 \leq y = \frac{m-u}{v-u} \\ v - \sqrt{(1-r)k_2} & \text{if} \quad y \leq r \leq 1 \end{cases}$$

$$\mu \approx (u+m+v)/3$$

$$\sigma^2 \approx [u(u-m) + v(v-u) + m(m-u)]/18$$

3 - Normal Distribution:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} \quad \text{for} \quad -\infty \leq x \leq \infty$$

$$\text{and} \quad \sigma > 0$$

r: is a random number from a standard normal distribution

$$x = \mu + \sigma r$$

4 - Exponential Distribution:

$$f(x) = \begin{cases} \frac{1}{\alpha} \exp(- x/\alpha) & \text{for} \quad x \geq 0 \quad \text{and} \quad \alpha > 0 \\ 0 & \text{otherwise} \end{cases}$$

r: is a random number from an exponential distribution with mean $\alpha > 0$.

$$x = r$$

$$\mu = \alpha$$

$$\sigma^2 = \alpha^2$$

5 - Gamma Distribution:

$$f(x) = \begin{cases} \dfrac{1}{\Gamma\alpha\beta^\alpha} x^{\alpha-1} e^{-x/\beta} & \text{if} \quad x \geq 0 \quad \text{and} \quad \alpha,\beta > 0 \\ 0 & \text{otherwise} \end{cases}$$

r: is a random number having the density $f(r) = \frac{1}{\Gamma\alpha} r^{\alpha-1} e^{-r}$ for $r \geq 0$

$$x = r\beta$$

$$\mu = \alpha\beta$$

$$\sigma^2 = \alpha\beta^2$$

6 - Beta Distribution:

$$f(x) = \begin{cases} \dfrac{\Gamma(\alpha+\beta)}{\Gamma\alpha\Gamma\beta(b-a)^{\alpha+\beta-1}} (x-a)^{\alpha-1}(b-x)^{\beta-1} & \text{for} \quad u \leq x \leq v \\ & \text{and} \quad \alpha,\beta > 0 \\ 0 & \text{otherwise} \end{cases}$$

r: random number having the density $f(r) = \frac{\Gamma(\alpha+\beta)}{\Gamma\alpha\Gamma\beta} (r)^{\alpha-1}(1-r)^{\beta-1}$ $0 \leq r \leq 1$

$$x = u + r(v-u)$$

$$\mu = u + \frac{\alpha}{\alpha+\beta} (v-u)$$

$$\sigma^2 = (v-u)^2 \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta-1)}$$

7 - Discrete Distributions: This category includes all discrete distributions and any customer oriented distribution. Its input consists of a finite set of ordered pairs, $\{(R_m, p(R_m))\}$. To generate a random number from any of these distributions, a random number $r \sim U(0,1)$ is generated, then the desired random realization is given by

$$x = P^{-1}(r).$$

Distribution	Index I	EX(I)	STDX(I)	VMIN(I)	VMAX(I)	IMSL Subroutine
Uniform	1	0.00	0.00	u	v	GGUBS
Triangular	2	m	0.00	u	v	GGUBS
Normal*	3	μ	σ	u	v	GGNML
Exponential*	4	α	0.00	u	v	GGEXN
Gamma*	5	α	β	u	v	GGAMR
Beta	6	α	β	u	v	GGBTR
Discrete	7	not applicable				GGUBS

Table 8

Input Parameters for SIMULT

~~In the Normal, Exponential and Gamma distributions, marked by * in~~ Table 8, a random value is accepted only if it is in the interval [u,v]; otherwise, it is rejected and a new random value is generated.

2 - Comparing the Approximate pdf with the "true" pdf: The "true" pdf of the project completion time is represented by the pdf obtained from the extensive Monte Carlo sampling denoted by F'(N); the cardinality of

F'(N) is NIN. This distribution is compared with the approximate pdf, denoted by F(N) which has NRR realizations. The objective of the comparison is to determine the maximum absolute deviation (MDV), the average value of the absolute deviations (ADV) between the two distributions, and the mean and the standard deviation of each distribution. The deviations are computed using linear interpolation. Figure 10 illustrates the concept of linear interpolation where the points generating the solid line represent the approximate pdf, F(N), and the scattered points in the plane (R, P(R)) represent the "true" pdf, F'(N).

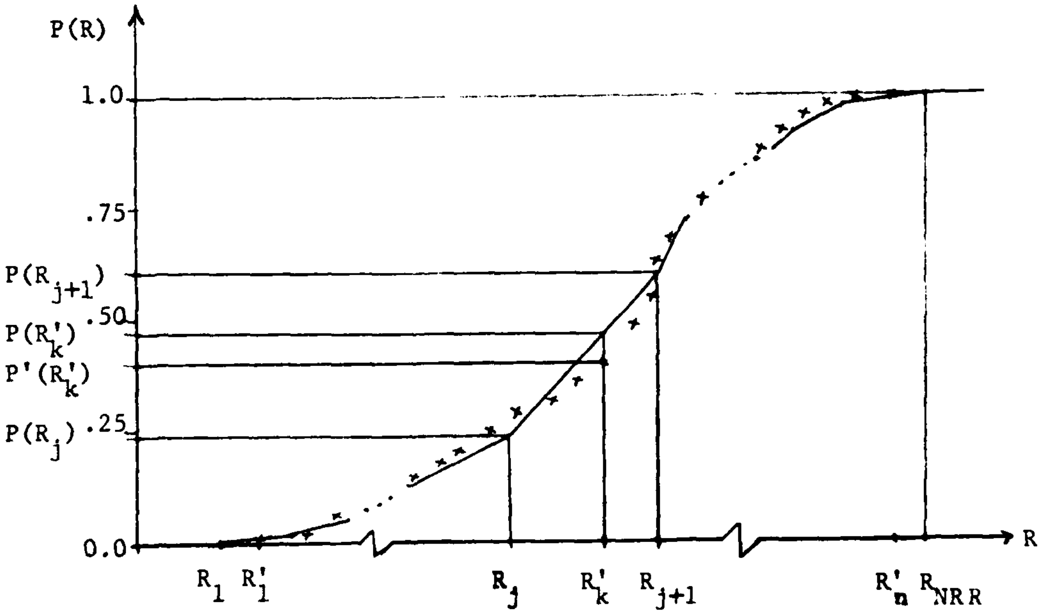


Figure 10

Linear Interpolation

If the number of realizations in $F'(N)$ is $n = NIN$, then we have n deviations. Let

$$D_k\text{: be the } k^{th} \text{ deviation where } k = 1,2,\ldots,n.$$

Hence,

$$MDV = \operatorname*{Max}_k\{|D_k|\}$$

and,

$$ADV = \sum_{k=1}^{n} |D_k|/n.$$

In the approximate pdf the minimum and maximum realization times of the project are always preserved; they are represented by R_1 and R_{NRR} respectively. Hence

$$R_1' \geq R_1$$

$$\text{and} \quad R_n' \leq R_{NRR}.$$

Therefore, for any R_k' of the true realizations, $k = 1,2,\ldots,n$, there exists an approximate realization R_j where $j < NRR$ such that

$$R_j \leq R_k' \leq R_{j+1}$$

Using this relation we can obtain the equation of the line segment connecting the two points $(R_j, P(R_j))$ and $(R_{j+1}, P(R_{j+1}))$. If such a line is denoted by

$$y = rx + s$$

where x is the realization axis and y is the probability axis, then the slope of the line is

$$r = (P(R_{j+1}) - P(R_j))/(R_{j+1} - R_j)$$

and the intercept is

$$s = P(R_j) - rR_j.$$

Hence, if R_k' is given,then using the line equation we calculate the corresponding approximate probability $P(R_k')$ where

$$P(R_k') = y = rR_k' + s,$$

then

$$D_k = P(R_k') - P'(R_k').$$

See Figure 10 for illustration.

The linear interpolation is carried out for all realizations except for those which lie in the 0.01 left and right tails of the sampled distribution. The exclusion of the two tails does not alter the values of MDV and may alter ADV only slightly, and speeds up the interpolation since many realizations with negligible probabilities might lie in the tails. For example, the application of the linear interpolation to Tables 6 and 7 led to the exclusion of the first four and last five realizations of Table 6. Table 8 has the complete output of the linear interpolation; the third column in the table headed by "APRXMTD PROB." represents $P(R_k')$ and the last column represents D_k. Figure 11 is a digital plot of the first three columns of Table 8. The symbol (·) represents the sampled distribution where (-) represents the approximate distribution and (x) is used whenever (·) and (-) are to be printed in the same location in the xy-plane.

I	REALIZATION	SAMPLED PROR.	APRXMTD PROB.	ACTUAL DIFFERENCE
1	19.6	.20000E-01	.14173E-01	-.58274E-02
2	20.5	.35000E-01	.23151E-01	-.11849E-01
3	21.3	.57000E-01	.38755E-01	-.18245E-01
4	22.2	.85000E-01	.63531E-01	-.21469E-01
5	23.0	.12500	.97532E-01	-.27468E-01
6	23.9	.18100	.16163	-.19373E-01
7	24.9	.23700	.23422	-.27838E-02
8	25.7	.31700	.30321	-.13793E-01
9	26.5	.40200	.38369	-.18304E-01
10	27.4	.48400	.47733	-.66729E-02
11	28.2	.56400	.56206	-.19395E-02
12	29.1	.64600	.64554	-.45729E-03
13	30.0	.72500	.71579	-.92102E-02
14	30.9	.79500	.78020	-.14793E-01
15	31.7	.85500	.83306	-.21939E-01
16	32.5	.88700	.88194	-.50564E-02
17	33.5	.92200	.92312	.11195E-02
18	34.2	.94800	.94687	-.11306E-02
19	35.0	.97000	.96652	-.34733E-02
20	36.1	.98000	.98372	.37264E-02
21	36.9	.98800	.98938	.13794E-02

The Average of the Absolute Values of the Deviations = .70003E-02
The Maximum of the Absolute Values of the Deviations = .27468E-01.It is No. 5
Number of Positive Deviates = 3
Number of Negative Deviates = 18

Table 8

Comparison of the Sampled and the Approximate
Probability Distribution Functions

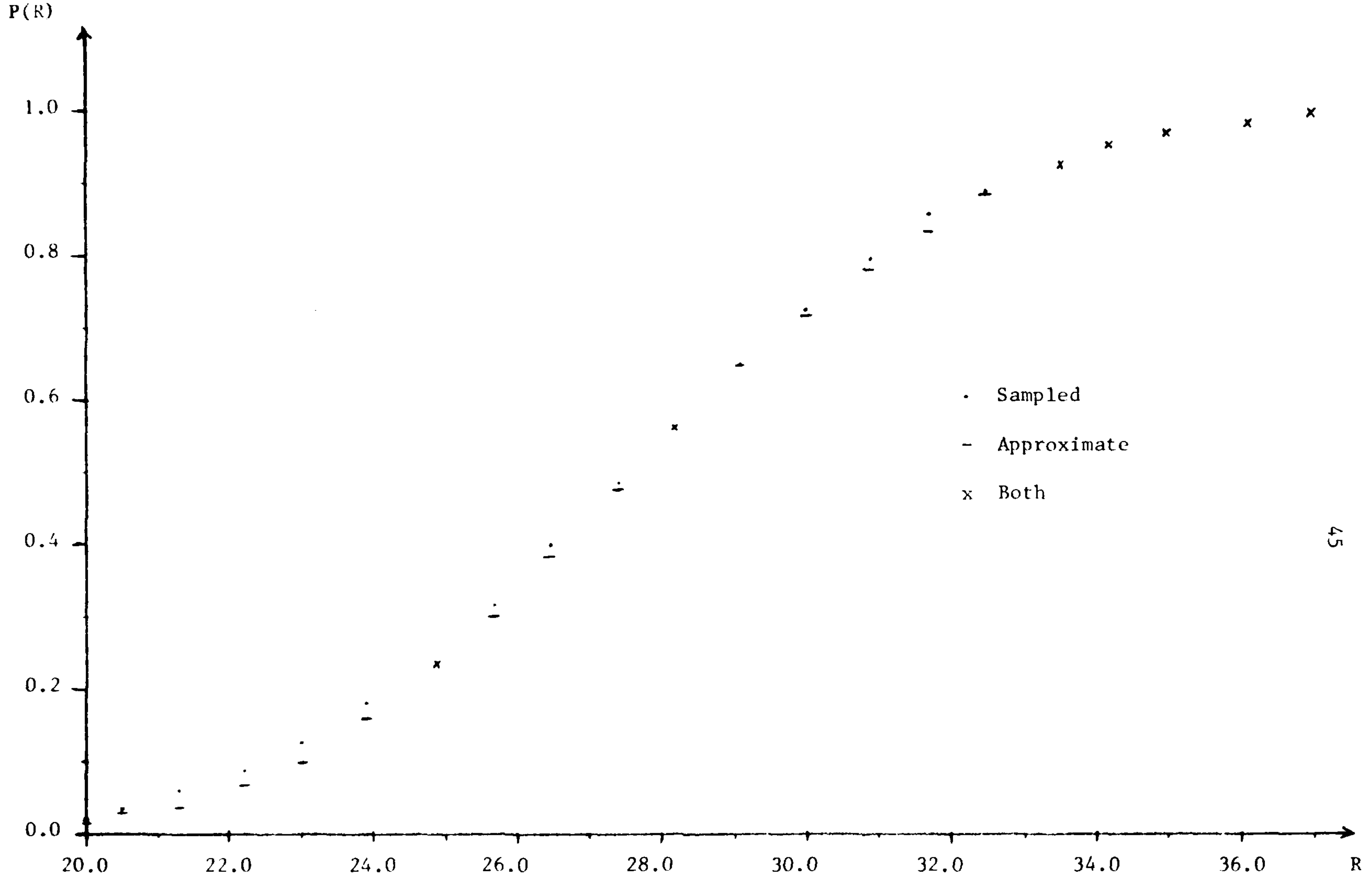


Figure 11

Comparison of the Sampled and the Approximate Probability Distribution Functions

VII. COMPUTATIONAL EXPERIENCE AND CONCLUSIONS

It is very difficult to judge the accuracy of the approximate pdf of the project completion time since the exact pdf is not known, (it may be known for small ANs), and the literature does not report other approximating procedures beside the various forms of MCS. Therefore, as it was clear from the previous section, we had to compare the approximate pdf with that obtained by MCS using the following four measures of performance:-

1 - Average value of the distribution

2 - Standard deviation

3 - The maximum of the absolute values of the deviations (MDV)

4 - The average of the absolute values of the deviations (ADV).

The variations in the measures of performance depend on the structure and size of the AN, the distributions of the activity times, the sample size in MCS, the accuracy of the discretization, and the values of NRR and NIN. In this section we discuss the impact of these factors on the measures of performance (MOP) and conclude the section with some conclusions concerning the approximating procedure.

Table 9 shows how the distribution type affects the MDP for a randomly generated AN with $N = 10$ and $|A| = 15$ where the sample size is set equal to 1000. The parameters of the distributions used are given in Table 10. In each of the eight problems considered in Table 9, the approximate average value of the project completion time is within 1% of the sampled average. The approximate average is slightly higher than the sampled average, while the approximate standard deviation is less than the sampled stan ard deviation. The graph of the density functions of each problem has the form

Problem	Distr. Type*	Comparison of the Approximate PDF with that of MCS					
		Average		S. Deviation		MDV	ADV
		APRX.	MCS	APRX.	MCS		
1	1	27.27	27.20	4.868	5.316	0.0426	0.0115
2	2	29.13	29.21	3.925	4.255	0.0513	0.0180
3	3	40.29	40.29	4.059	4.180	0.0585	0.0206
4	4	12.50	12.49	3.511	3.899	0.0328	0.00992
5	5	16.85	16.61	3.504	3.401	0.0436	0.0122
6	6	36.96	37.16	3.879	4.194	0.0772	0.0275
7	7	18.50	18.51	2.055	2.076	0.0083	0.0016
8	All	28.46	28.07	3.808	4.005	0.0274	0.0070

*For the specification of each distribution see Table 10 below.

Table 9

Sensitivity of the Approximation
Procedure to the PDF's

Order	Dist. Type	EX	STDX	VMIN	VMAX
1	Uniform	5.0	-	0.00	10.00
2	Triangular	5.0	-	1.00	11.00
3	Normal	8.0	2.0	2.00	14.00
4	Exponential	2.0	2.0	0.00	15.00
5	Gamma	3.0	1.0	0.00	10.00
6	Beta	3.0	2.0	1.00	11.00
7	Discrete	{(2.0,0.20),(3.0,0.30),(4.0,0.30),(5.0,0.20)}			

Table 10

Probability Distribution Functions
Used in the Analysis

shown in Figure 12. The sampled graph converges to the approximate graph as the sample size increases; Table 5 of Section VI gives an example of such convergence. The values of MDV and ADV vary with the time distributions adopted; but in the eight problems of Table 9 MDV is always less than 0.08 and ADV is less than 0.03. The smallest values for MDV and ADV are obtained in problem 7 where a discrete distribution was used, the second smallest values of MDV and ADV were obtained in problem 8 where some of the activities in problem 8 have discrete distribution. This is expected since the errors of discretization in problem 7 do not exist and in problem 8 they are less than in the remaining problems. The accuracy of the approximation can be enhanced by having more accurate discretization; this was the case in problem 4 where the exponential distribution was approximated by thirty points, while each of the other continuous distributions was approximated by only twenty points.

In Table 11 the uniform distribution and a sample of size 1000 are used to examine the effects of the AN size on the MOP. Both the MDV and ADV increase as the AN size increases; perhaps such an increase is due to retaining the sample size constant since large ANs require larger sample sizes. The graphs of the distributions of the problems in Tables 9 and 11 have the general form of Figures 12 and 13; which agree with the general forms obtained by Van Slyke [8] in sampling different PERT networks. The measure of MDV, in most of the problems considered, has its value from within the values of the first 30% of the distribution. The parameter D_k used in Section VI.2 where

$$D_k = P(R_k') - P'(R_k')$$

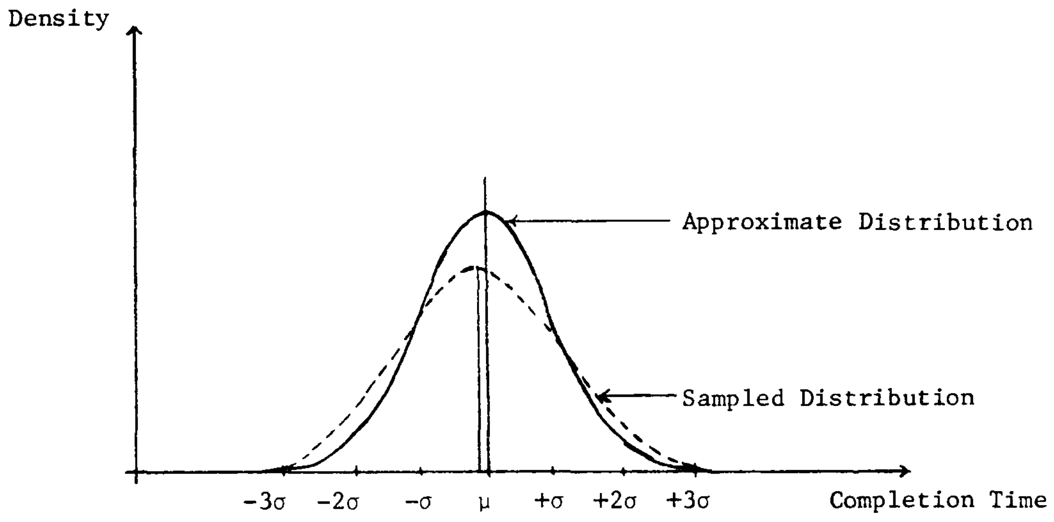


Figure 12

General Forms of the Probability Density Functions of the Project Completion Time

Problem	Nodes (N)	Arcs (A)	Comparison of the Approximated PDF with that of MCS					
			Average		S. Deviation		MDV	ADV
			APRX.	MCS	APRX.	MCS		
1	10	15	27.27	27.20	4.868	5.316	0.0426	0.0115
2	20	40	47.37	46.47	6.733	7.625	0.0557	0.0162
3	30	50	52.30	51.89	6.251	7.139	0.0525	0.0169
4	40	60	58.98	57.71	6.962	8.174	0.0633	0.01816
5	40	80	56.06	55.14	5.952	6.553	0.0303	0.01147
6	50	75	62.54	62.58	7.698	8.224	0.0651	0.0259
7	50	100	67.38	65.56	6.182	7.752	0.0880	0.0263
8	60	150	82.82	80.05	7.155	9.074	0.1082	0.0306

Table 11

Computational Experience for Different Size AN's with Uniform Distribution

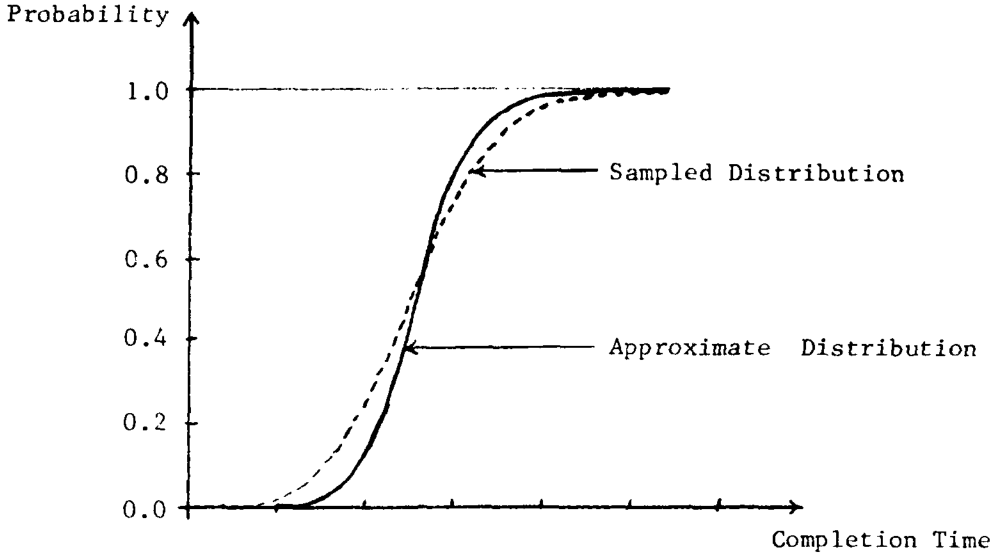


Figure 13

General Form of the Sampled and the Approximate Distributions

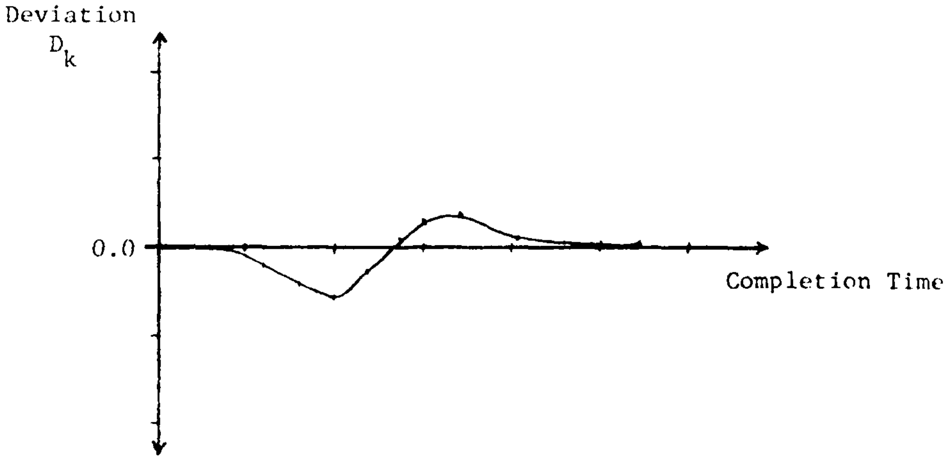


Figure 14

The Behavior of the Deviation D_k

tends to have negative values in the first half of the distribution and positive values in the second half. As the errors in the discretization decrease and the sample size in MCS increases the curve of D_k, having the general form of Figure 14, converges toward the abscissa.

The CPU time requirements of the approximating procedure excluding the MCS time are minimal. It is always less than half a minute for an AN of size $(N,A) \leq (60,200)$ with Uniform distribution on AMDAHL V-7. The CPU time requirements for MCS with a fixed sample size depend on the size of the AN and on the type of pdf's used. For a sample of size 1000 for the problem G(60,150) with a Uniform distribution the CPU time was about two minutes; the CPU time may double or triple if other distributions, such as the Normal or Beta, are used.

From the preceeding discussion we conclude the following:-

1 - The approximation is at its best if the activities have discrete distributions to start with. The accuracy of the approximation can be improved by reducing the errors of discretization.

2 - The distribution of the project completion time approaches normality regardless of the type of the distributions used at the outset. This was the case in all the problems tested. The approximated mean and standard deviation of the distribution are very close to the "true" mean and standard deviation. In fact it is bounded from below by the best known estimate, which is developed by Elmaghraby [2], and bounded from above by the true mean.

3 - In comparison with the pdf obtained by MCS, the sampled distribution converges toward the approximate pdf as the sample size increases.

4 - The maximum value of the absolute deviation, MDV, is within the first 30% of the distribution; this observation increases the applicability of the approximate pdf since the realizations of the major interest are those on the right half or tail of the pdf.

5 - The processing time requirements of the approximating procedure excluding the sampling time are minimal. It is always less than half a minute for any AN of size $(N,A) \leq (60,200)$ on AMDAHL V-7.

VIII. REFERENCES

1. Clark, Charles E., (1961), "The Greatest of a Finite Set of Random Variables", Oper. Res. 9(2), 146-162.

2. Elmaghraby, S. E., Activity Networks: Project Planning and Control by Network Models, 1977, John Wiley & Sons, Inc.

3. Feller, W., Introduction to Probability Theory and Its Applications, Vol. 1, 1968, John Wiley & Sons, Inc.

4. Hamming, R. Wesley, Numerical Methods for Scientists and Engineers, 1962, McGraw-Hill.

5. Herroelen, W. and Caestecke, G., "The Generation of Random Activity Networks", March 1979, Report No. 7906, Department of Applied Economic-Sciences, Katholieke Universiteit, Leuven, Belgium.

6. International Mathematical and Statistical Library (IMSL), IMSL Inc., GNB Building, Houston, Texas.

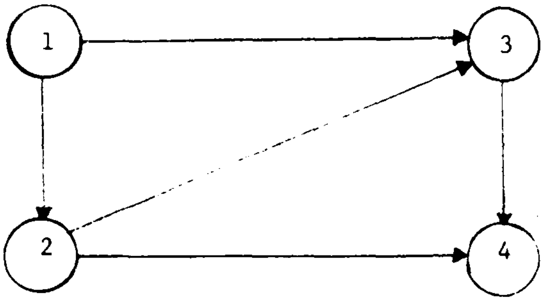
7. Singleton, R., "On Computing the Fast Fourier Transformation", Communications of the ACM, Vol. 10, No. 10, October 1967.

8. Van Slyke, R. M., "Monte Carlo Methods and the PERT Problem", Oper. Res. Vol. 11(5), 839-860.

APPENDIX A

RANDOM NETWORK GENERATOR

The nodes in G(N,A) are numbered such that an arc always leads from a small number to a larger one, and there is only one start and one end node to the AN. An immediate consequence of such a numbering scheme is that the adjacency matrix is always upper triangular with zero diagonal. A typical AN and adjacency matrix is given in Figure A.1 below for N = 4 and $|A| = 5$.



$$A = [a_{ij}] = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 0 \\ 2 & & 0 & 1 & 1 \\ 3 & & & 0 & 1 \\ 4 & & & & 0 \end{array}$$

Table A.1

An Activity Network and Its Adjacency Matrix

Another consequence is that for a given N and |A| several feasible G(N,A) may be generated. Figure A.2 lists the other (three) alternative AN types for N = 4 and |A| = 5

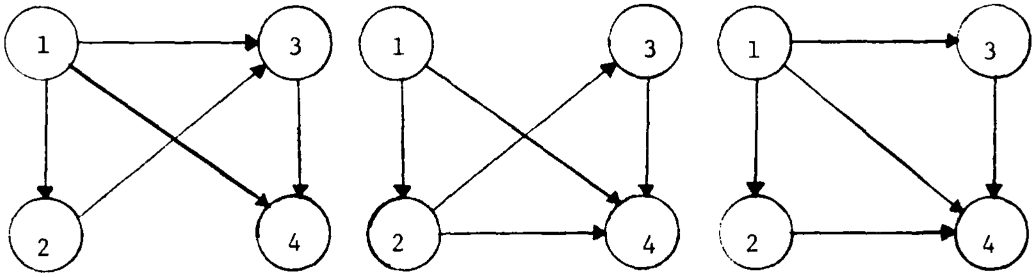


Figure A.2

The Remaining Feasible ANs with N = 4, |A| = 5

Consequently, the random generation of a G(N,A) for a fixed N and |A| implies that the resultant network types should have equal probabilities of occurrence. In general, the following two procedures should be able to satisfy this requirement. The first method denoted by the "Deletion Method" starts from the completely connected AN; i.e., the adjacency matrix filled with ones, and deletes the necessary number of arcs until |A| arcs are left. This is done by substituting zeroes for ones until |A| ones are left in the adjacency matrix. The second procedure, denoted by the "Addition Method", starts with the unordered AN; i.e., the adjacency matrix filled with zeroes, and generates the required number of arcs |A|. This is done by substituting |A| ones for zeroes in the adjacency matrix. In the following we discuss the rationale of both methods.

1 - Deletion Method: Let $\bar{A} = (a_{ij})$ be the adjacency matrix corresponding to a completely connected network. Figure A.2 gives an example for $N = 4$ and $|A| = 5$. Then let

$$n_i = \sum_j a_{ij} = N - i \text{ (outdegree) ,} \qquad \text{(A.1)}$$

and for each j let

$$m_j = \sum_i a_{ij} = j - 1 \text{ (indegree) .}$$

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 1 \\ 2 & & 0 & 1 & 1 \\ 3 & & & 0 & 1 \\ 4 & & & & 0 \end{array}$$

Figure A.2

Completely Connected AN

The Deletion Method now reduces to the random deletion of

$$N(N-1)/2 - |A|$$

ones in the adjacency matrix, such that

$$n_i \geq 1 \quad \text{for all} \quad i \neq N \quad \text{and} \quad n_N = 0 \qquad \text{(A.2)}$$

and

$$m_j \geq 1 \quad \text{for all} \quad j \neq 1 \quad \text{and} \quad m_1 = 0 \qquad \text{(A.3)}$$

For any AN, the above conditions simply state that at least one arc must leave every node except the last and at least one arc should enter every node except the first.

The Deletion Method should generate activity networks with equal probabilities for the different feasible network types, i.e., all existing ones in the adjacency matrix for the completely connected network should receive equal deletion probabilities given the above-mentioned consistency constraints. This can be achieved by numbering all the ones in the adjacency matrix for the completely connected AN from left to right and consecutively in the rows, as illustrated in Figure A.4.

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1(1) & 1(2) & 1(3) \\ 2 & & 0 & 1_{(4)} & 1(5) \\ 3 & & & 0 & 1(6) \\ 4 & & & & 0 \end{array}$$

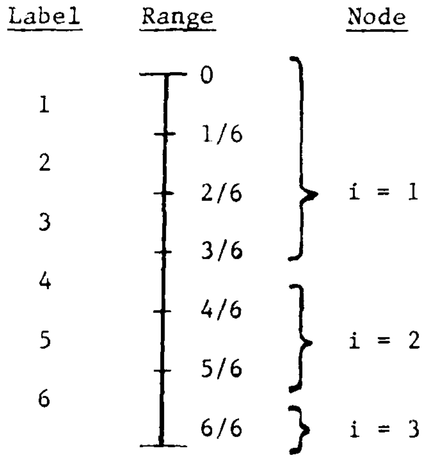


Figure A.4

Label and Probability Assignment

The corresponding numbers are then assigned to equal intervals in the range of a uniformly distributed variable. Then drawing a random number yields an interval which, in turn, identifies the label of a corresponding arc. The interval corresponding to a node i^* has a length equal to $(N-i^*)$ times the interval length of a label. For example in Figure A.4 the interval corresponding to node $i^* = 2$ has a length of

$$(4-2)(1/6) = 1/3.$$

It can also be seen from Figure A.4 that $i^* = 2$ is preceded by 3 intervals, each is of length 1/6; i.e., in general, node i* is preceded by at least

$$\sum_{0<i<i^*} (N-i) = (i^*-1)N - i^*(i^*-1)/2 \qquad (A.4)$$

labeled intervals.

In order to generate an i^*, let $Y \sim U(0,1)$ and let

$$X = Y \cdot N(N-1)/2 \qquad (A.5)$$

where N(N-1)/2 denotes the total number of labels (total number of arcs in the AN). Now the interval relation between X and i^* implies that (see Eq. (A.4))

$$X \geq (i^*-1)N - i^*(i^*-1)/2$$

or with $\alpha \geq 0$ we have

$$i^{*2}/2 - (N+1/2)i^* + (N+X-\alpha) = 0 \quad ,$$

which yields

$$i^* = (N+1/2) \pm \sqrt{(N+1/2)^2 - 2(N+X-\alpha)} \quad . \qquad (A.6)$$

Since $i^* \leq N - 1$ we must select the "-" root. Moreover, since $\alpha \geq 0$, Eq. (A.6) reduces to

$$i^* \leq (N+1/2) - \sqrt{(N+1/2)^2 - 2N - 2X} \quad ,$$

or

$$i^* \leq (N+1/2) - \sqrt{(N-1/2)^2 - 2X} \quad .$$

Substituting from Eq. (A.5) yields

$$i^* \le (N+1/2) - \sqrt{N(N-1)(1-Y) + 1/4}$$

Also by a symmetrical argument that considers the nodes that are larger than i^*, we have

$$X \le i^*N - i^*(i^*+1)/2$$

we find

$$i^* \ge (N-1/2) - \sqrt{N(N-1)(1-Y) + 1/4}.$$

But $Y \sim U(0,1)$ implies that $(1-Y) \sim U(0,1)$; hence let $\beta = \sqrt{N(N-1)Y + 1/4}$, then

$$N - 1/2 - \beta \le i^* \le N + 1/2 - \beta$$

or

$$i^* = \lfloor N + 1/2 - \beta \rfloor \quad \ldots . \tag{A.7}$$

Given this value of i, we draw a new random observation of $Y \sim U(0,1)$ and rescale into $X \sim U(i+1, N+1)$ by setting

$$X = Y(N-i^*) + i^* + 1$$

which in turn yields

$$j^* = \lfloor i^* + 1 + Y * (N-i^*) \rfloor . \tag{A.8}$$

The corresponding arc (i^*, j^*) is deleted from the AN provided that conditions A.2 and A.3 are satisfied. This procedure is repeated until

$$\sum_i n_i = \sum_j m_j = |A| .$$

2 - The Addition Method: The Deletion Method will delete N(N-1)/2 - $|A|$ arcs. For certain values of N and $|A|$, this may be a time consuming process. Suppose we have to generate a network with N = 4 and $|A|$ = 5, then the deletion method will have to delete 1 arc; however, if N = 100 and A = 150, then 4800 out of 4950 arcs need to be deleted.

Under such conditions, the Addition Method may prove to be less time-consuming. As a consequence of the node labeling procedure adopted, there should always be an arc connecting nodes 1 and 2 and an arc connecting nodes N - 1 and N. Consequently, we believe the Deletion Method is to be preferred if

$$|A| \geq N(N-1)/4 + 1,$$

and we prefer the Addition Method if otherwise.

Consider now the previous example with n = 4 and $|A|$ = 5. Figure A.5 represents the initial adjacency matrix and the corresponding network. It can be observed from Figure A.5 that node 2 is not yet an emitting node and

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 0 & 0 \\ 2 & & 0 & 0 & 0 \\ 3 & & & 0 & 1 \\ 4 & & & & 0 \end{array}$$

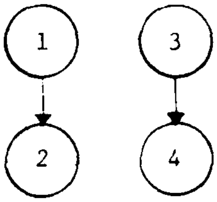


Figure A.5

Initial Network in the Addition Method

node 3 is not yet a receiving node. In general the initial network will be characterized by n = N - 3 non-emitting nodes, and m = N - 3 non-receiving nodes. This means that

$$f = |A| - 2 - m - n$$

of the remaining arcs may be inserted arbitrarily; in our example

$$f = 5 - 2 - 1 - 1 = 1$$

arc of the three remaining may be generated completely arbitrarily (i.e., both of its terminal nodes are arbitrary), since at least an arc must enter node 3, and one arc must leave node 2. Hence, the terminal of the arc from node 2 and the origin of the arc to node N-1 may be selected arbitrarily. Consequently, the Addition Method will start from the initial network and adjacency matrix (all $a_{ij} = 0$ except $a_{12} = 1$ and $a_{N-1,N} = 1$). It uses formulas (A.7) and (A.8) to generate an arc as long as the residual free arcs f is > 0 where

$$f = A - \ell - m - n > 0$$

and initially, the number of generated arcs $\ell = 2$, the non-emitting nodes $n = N - 3$,and the number of non-receiving nodes $m = N - 3$. Each time an arc is generated in this manner, the adjacency matrix is updated, the value of ℓ is set to

$$\ell = \ell + 1,$$

and depending on the outcome either

$$m = m - 1$$

and/or

$$n = n - 1.$$

The Addition Method developed by Herroelen and Caestecke [5] is represented by Flowchart 1. If $\ell < A$ and $f = 0$ we check if $m = 0$. If $m \neq 0$, indicating that there is at least one non-receiving node, we locate the column j* in the adjacency matrix that is completely filled with zeroes (if ties develop, take the highest column index). We generate a corresponding i* using

$$i^* = \lfloor 1 + (j^* - 1)Y \rfloor \quad \text{where} \quad Y \sim U(0,1).$$

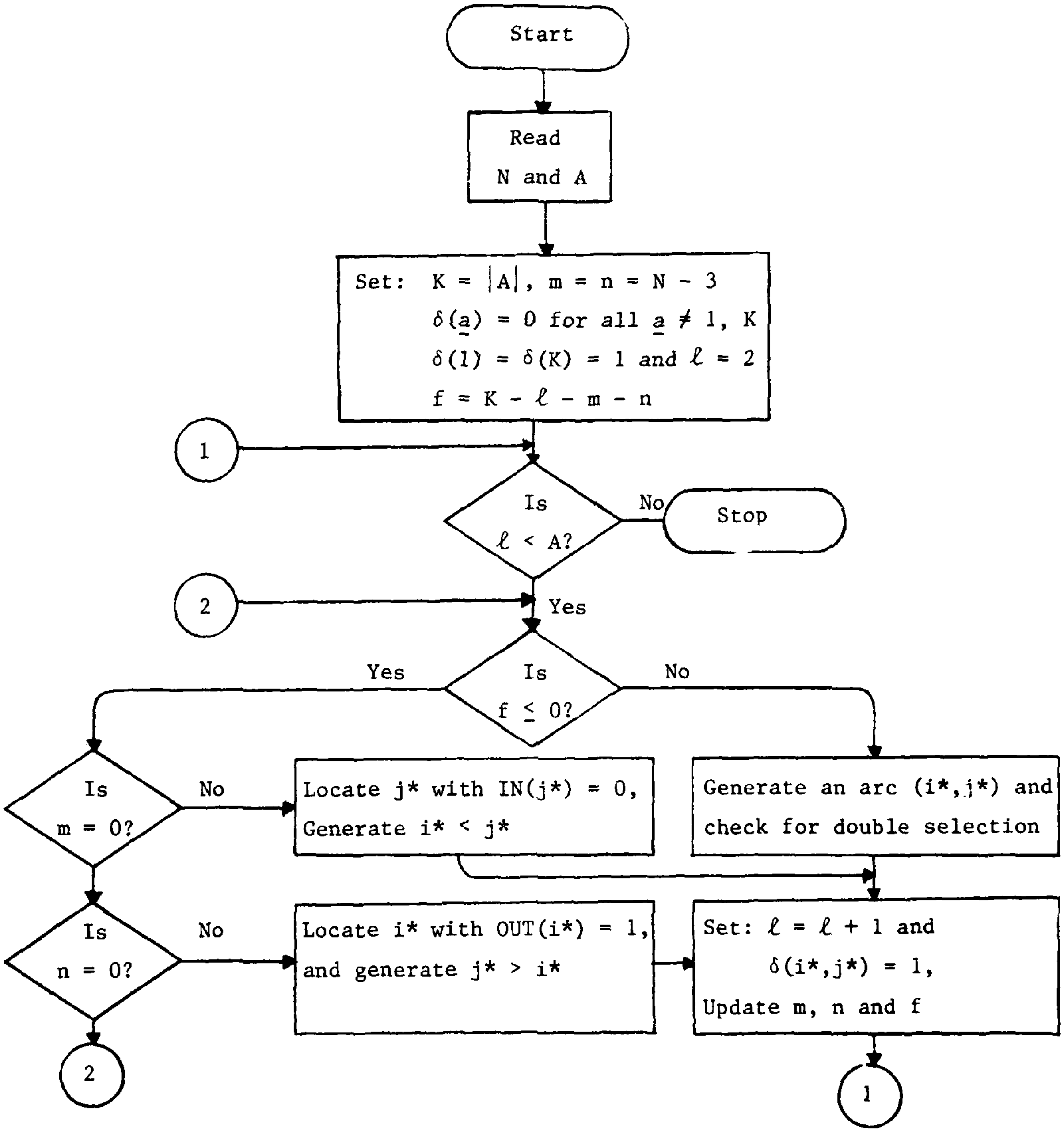
Update the adjacency matrix, and the values of ℓ, m, n and f then continue until either $\ell = A$ where the process stops, or $\ell < A$ and $f = m = 0$; in such a case we check if $n = 0$. If $n \neq 0$, then there is at least one non-emitting node. We locate any zero row, $i^* < N - 1$, in the adjacency matrix, and generate a j* using the formula

$$j^* = \lfloor i^* + 1 + (N - i^*)Y \rfloor \quad \text{where} \quad Y \sim U(0,1).$$

We continue until either $\ell = A$ or $n = 0$, where in either case the process stops.

Ideally as soon as $m = 0$ and $n = 0$ we should have $\ell = |A|$. However the Addition Method presented in Flowchart 1 does not guarantee this result. If $|A| < 2N - 4$ then the Addition Method may fail to generate a feasible AN, and if the feasibility conditions are imposed then the method may generate more arcs than what is required. The following example illustrates this defficiency:

Example: Let $N = 10$ and $|A| = 12$ then Table 1 below summarizes the steps taken by the Addition Method represented by Flowchart 1. It is obvious that at step 10 we have $\ell = |A| = 12$ and according to Flowchart 1 the process stops, even though there are nodes not connected from above,


Start
Read
N and A
Set: K = |A|, m = n = N - 3
δ(a) = 0 for all a ≠ 1, K
δ(1) = δ(K) = 1 and ℓ = 2
f = K - ℓ - m - n
1
Is
ℓ < A?
No
Stop
2
Yes
Yes
Is
f ≤ 0?
No
Is
m = 0?
No
Locate j* with IN(j*) = 0,
Generate i* < j*
Generate an arc (i*,j*) and
check for double selection
Is
n = 0?
No
Locate i* with OUT(i*) = 1,
and generate j* > i*
Set: ℓ = ℓ + 1 and
δ(i*,j*) = 1,
Update m, n and f
2
1

Flowchart 1

The Addition Method

i.e., there are non-emitting nodes. If at step 10 we let the process to go to ② instead of ① in Flowchart 1, then the process will terminate only after changing the, "go to 2" in the decision "Is n = 0?" to stop. In such a case the process stops after n reaches zero, where the number of generated arcs can exceed 12. In fact in this example the Addition Method can generate up to 16 arcs. For the realization presented in Table 1, ℓ goes up to 15 arcs.

This deficiency is always possible for all $|A| \epsilon [N - 1, 2N - 5]$. For $|A| \geq 2N - 4$ the Addition Method as outlined in Flowchart 1 appears to be working. In section II we modify the Addition Method to avoid this deficiency.

Step k	ℓ	m	n	$f = A - \ell - m - n$
0	2	7	7	$12 - (2 + 7 + 7) = - 4$
1	3	6	6	$12 - (3 + 6 + 6) = - 3$
2	4	5	6	- 3
3	5	4	6	- 3
4	6	3	6	- 3
5	7	2	6	- 3
6	8	1	6	- 3
7	9	0	6	- 3
8	10	0	5	- 3
9	11	0	4	- 3
10	12	0	3	- 3
11	13	0	2	- 3
12	14	0	1	- 3
13	15	0	0	- 3

1	2	3	4	5	6	7	8	9	10
	1^0	0	0	1^5	0	1^3	1^2	0	0
		1^7	1^6	0	1^4	0	0	1^1	0
			0	0	0	0	1^8	0	0
				0	0	0	0	0	1^9
					0	0	0	0	1^{10}
						0	0	0	0
							0	0	0
								0	0
									1^0

$$\text{Step number} \rightarrow k \qquad \underbrace{1 \text{ or } 0}_{\text{Arc indicator}}$$

entry (i,j)

According to Flowchart 1 the process stops here((step 10).

If the process tops here then $\ell > A$

Table 1

One Possible Realization of Flowchart 1

APPENDIX B

THE 2m METHOD

It was mentioned in Section III that the first method of discretizing a continuous pdf is to use the first 2m moments of the continuous distribution to solve the following system of nonlinear equations

$$\sum_{k=1}^{m} x_k^n p(x_k) = E(x^n) = e_n \quad \text{for} \quad n = 0,1,2,\ldots,2m-1 \tag{1}$$

In a matrix form we have

$$VP = E.$$

The following two methods have been tried to solve this system of nonlinear equations, but neither system succeeded in solving it for m > 8. These methods are:

1 - Brown Method: which is documented in IMSL [6] library under the name ZSYSTM. Starting with an initial solution ZSYSTM is supposed to converge to a solution within ε from a feasible solution. However, many runs to different values of m and different initial solutions proved that ZSYSTM was not converging, and often terminated because of a singularity that occurred in the iterations, due mainly to the nature of Vendermonde matrix V. Two other packages SBROWN and SNGINT developed by the Argonne National Laboratory have been tried; neither succeeded in solving the above system. This failure led to the search for other methods. The following method was successful in solving the above system, but only for small value of m ($\leq$ 8).

2 - Gaussian Quadrature [4]: To solve the above system of nonlinear equations the procedure is as follows:

(i) Determine the sample polynomial

$$\pi(x) = \sum_{k=0}^{m} c_k x^k,$$

The coefficients $\{c_k\}$ are determined uniquely using the following system of linear equations after setting $c_m = 1$.

$$\begin{gathered} c_o e_o + c_1 e_1 + c_2 e_2 + \ldots + c_{m-1} e_{m-1} + e_m = 0 \\ c_o e_1 + c_1 e_2 + c_2 e_3 + \ldots + c_{m-1} e_m + e_{m+1} = 0 \\ c_o e_2 + c_1 e_3 + c_2 e_4 + \ldots + c_{m-1} e_{m+1} + e_{m+2} = 0 \\ \vdots \\ c_o e_{m-1} + c_1 e_m + c_2 e_{m+1} + \ldots + c_{m-1} e_{2m-2} + e_{2m-1} = 0 \end{gathered}$$

(ii) The set of realizations (discrete points) are determined by solving the polynomial

$$\sum_{k=0}^{m} c_k x^k = 0$$

where all m solutions are simple and real (since $0 \leq u \leq x \leq v$).

(iii) The corresponding probabilities are determined by substituting for x_k in the first m equations of (1) then solve uniquely for $p(x_k)$.

This algorithm was programmed and tested; it works for $m \leq 8$. It is not recommended for discretization since it is very sensitive to the values of $E(x^n)$, and requires the solution of two systems each of m linear equations, and the solution of a polynomial of the m^{th} degree, each time it is used to approximate a distribution. Furthermore, the user would never know when the procedure will "blow-up".

APPENDIX C

THE INPUT FORMAT

The approximating procedure has been programmed and for the ease and flexibility of its use, the program is in two parts: the first is used when the AN is given and the second is used if the AN is to be generated. This section describes the input requirements of each part.

1 - Input for an available AN: The input data is listed according to the following order and format.

(a) Control Card: It is the first input card; it contains the control parameters listed in the following order according to the format (F5.3, 7I5);

SCAL,N,M,NRR,NCONT,MCS,NSIM,KEY

where

SCAL = Δ, the interval width (mesh) used in DISCRT.

N = Number of nodes

M = $|A|$, number of arcs

NRR = Number of desired ordered pairs in the approximated distribution.

NCONT = $\begin{cases} 0 \text{ if the AN has no arcs with continuous distribution} \\ 1 \text{ otherwise} \end{cases}$

MCS = $\begin{cases} 0 \text{ if the Monte Carlo sampling is not desired} \\ 1 \text{ otherwise} \end{cases}$

NSIM = Number of samples if MCS = 1

KEY = Number of milestones (key nodes).

(b) List of the KEY nodes: These are listed in a non-decreasing order according to the format (16I5). They are integers denoted by the symbol KEYN.

(c) Identity of the Activities: This consists of four integer values listed on one card, according to the format 4I5, for each activity. These values are:

NS,NE,NDSTT,NR

where

NS: starting node

NE: end node

NDSTT = 1,2,...,7, indicator of the activity pdf.

NR: number of ordered pairs of distribution if NDSTT = 7.

(d) Distribution Parameters: Those are read, in the order of the arcs (which is the topological order of the AN). If NDSTT($\underline{a}$) ≠ 7, i.e., the arc has a continuous distribution, then the following four values listed according to the format (4F10.4) are needed for each activity. These are:

EX,STDX,VMIN,VMAX

which are explained in Table 8 of Section VI.

If NDSTT($\underline{a}$) = 7 then NR($\underline{a}$) ordered pairs, $\{(R(k), p(R(k)))\}$, are listed according to the format (4(F10.2, F10.4)), hence each card contains four ordered pairs.

NOTE: The (d) part of the input assumes each activity has its own distribution. The input routine can be changed to accommodate the assignment of a given distribution to a set of activities.

2 - Input if the AN is to be Generated: The input for this part is limited to the following segments:

(a) Control Card: as in 1(a) above.

(b) List of the KEY nodes: as in 1(b) above.

(c) Activity-Distribution Assignment: This is a vector of length NOI, where NOI is the number of intervals (partitions) of the set A, where each partition has one p.d.f. Each entry in this vector consists of

NULT,NDS,NT

where

NULT: number of the activity representing the upper limit of the partition

NDS: 1,2,...,7, indicator of the pdf of the partition

NT: number of ordered pairs of the pdf if NDS = 7

This vector is listed according to the format (16I5). (Notice that the first entry at the beginning of the vector is the value of NOI).

(d) Distribution Parameters: as in 1(d) above.

At the end of both parts of the input data we add the information needed for the digital plotter, the plotter is USPLT of the IMSL Library. It is listed on three cards as follows:

1 - First card,which has the format (4F10.2, 10A1, 4I5),contains the following:

(a) RAN(I) for I = 1,2,3,4: Four values specifying the minimum and maximum of the x and y axis respectively. If RAN(I) are set equal to zero, then the program determines the x and y ranges.

(b) PCH: Contains up to 10 plot characters, one for each function. If not specified leave PCH(1) and PCH(2) blank.

(c) IOP: A zero one input parameter indicating number of printer columns available.

(d) INC: Displacement between values in x to be considered.

(e) IY: First dimension of the arry y.

(f) NF: Number of functions to be plotted.

2 - Second card contains 72 characters of title information.

3 - Third card contains 36 characters for each axis for its annotation.

ATE

LME